

5 Data content and structure

5.1 Application schemas – Overview

5.1.1 Application schemas included in the IRs

Articles 3, 4 and 5 of the Implementing Rules lay down the requirements for the content and structure of the data sets related to the INSPIRE Annex themes.

IR Requirement

Article 4

Types for the Exchange and Classification of Spatial Objects

1. For the exchange and classification of spatial objects from data sets meeting the conditions laid down in Article 4 of Directive 2007/2/EC, Member States shall use the spatial object types and associated data types, enumerations and code lists that are defined in Annexes II, III and IV for the themes the data sets relate to.
2. Spatial object types and data types shall comply with the definitions and constraints and include the attributes and association roles set out in the Annexes.
3. The enumerations and code lists used in attributes or association roles of spatial object types or data types shall comply with the definitions and include the values set out in Annex II. The enumeration and code list values are uniquely identified by language-neutral mnemonic codes for computers. The values may also include a language-specific name to be used for human interaction.

The types to be used for the exchange and classification of spatial objects from data sets related to the spatial data theme *Buildings* are defined in the following application schemas (see following sections):

- BuildingsBase application schema describes the concepts that are common to all other *Buildings* application schemas; it contains mainly the **core normative semantics** of theme *Buildings*
- Buildings2D application schema describes the **2D geometric representation** of the spatial object types defined in Buildings Base application schema, namely buildings and building parts; it inherits from the common semantics of Buildings base
- Buildings3D application schema describes the **3D geometric representation** of the spatial object types defined in Buildings Base application schema, namely buildings and building parts; it inherits from the common semantics of Buildings base

The application schemas specify requirements on the properties of each spatial object including its multiplicity, domain of valid values, constraints, etc.

NOTE The application schemas presented in this section contain some additional information that is not included in the Implementing Rules, in particular multiplicities of attributes and association roles.

TG Requirement 1 Spatial object types and data types shall comply with the multiplicities defined for the attributes and association roles in this section.

An application schema may include references (e.g. in attributes or inheritance relationships) to common types or types defined in other spatial data themes. These types can be found in a sub-

section called “Imported Types” at the end of each application schema section. The common types referred to from application schemas included in the IRs are addressed in Article 3.

IR Requirement

Article 3

Common Types

Types that are common to several of the themes listed in Annexes I, II and III to Directive 2007/2/EC shall conform to the definitions and constraints and include the attributes and association roles set out in Annex I.

NOTE Since the IRs contain the types for all INSPIRE spatial data themes in one document, Article 3 does not explicitly refer to types defined in other spatial data themes, but only to types defined in external data models.

Common types are described in detail in the Generic Conceptual Model [DS-D2.7], in the relevant international standards (e.g. of the ISO 19100 series) or in the documents on the common INSPIRE models [DS-D2.10.x]. For detailed descriptions of types defined in other spatial data themes, see the corresponding Data Specification TG document [DS-D2.8.x].

5.1.2 Additional recommended application schemas

In addition to the application schemas listed above, the following additional application schemas have been defined for the theme *Buildings* (see following sections):

- BuildingsExtendedBase application schema describes the **additional semantics** that should be used to extend normative profiles, whatever the chosen geometric representation (2D or 3D) is.
- BuildingsExtended2D application schema describes the **2D geometric representation** of the additional spatial object types (namely installations, other constructions, building units); it inherits both from the common semantics of <Buildings ExtendedBase> and of the 2D geometric representation of buildings and building parts.
- BuildingsExtended3D application schema describes both the **3D geometric representation** of the additional spatial object types (namely installations, other constructions, building units) and the additional concepts that should be used to provide more detailed information about buildings and associated objects, when represented by 3D data (walls, roofs, openings, room, textures, ...) ; it inherits both from the common semantics of <Buildings ExtendedBase> and of the 3D geometric representation of buildings and building parts.

These additional application schemas are not included in the IRs. They typically address requirements from specific (groups of) use cases and/or may be used to provide additional information. They are included in this specification in order to improve interoperability also for these additional aspects and to illustrate the extensibility of the application schemas included in the IRs.

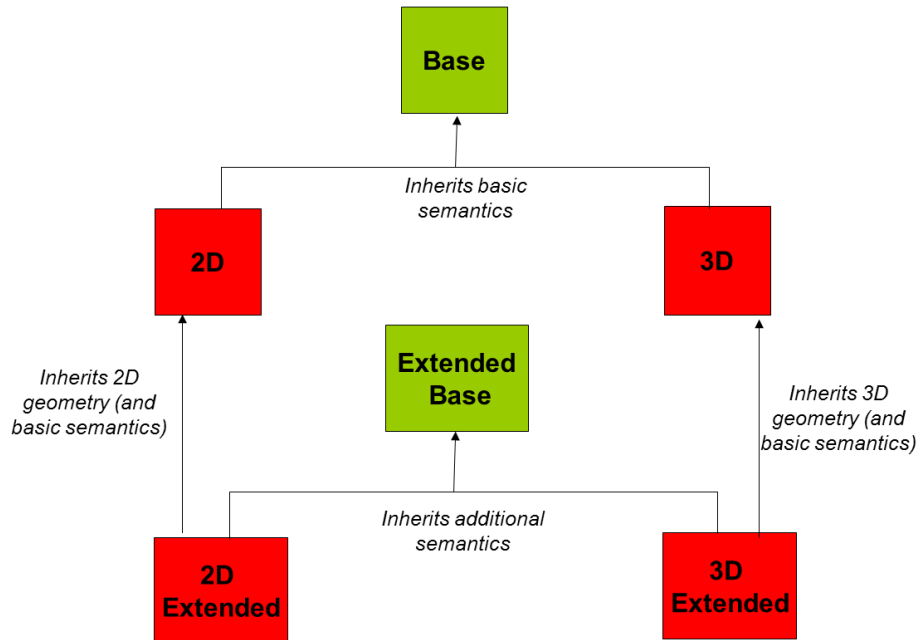


Figure 11: Dependencies between application schemas of theme Buildings

Recommendation 1 Additional and/or use case-specific information related to the theme *Buildings* should be made available using the spatial object types and data types specified in the following application schema(s): BuildingsExtendedBase, BuildingsExtended2D, BuildingsExtended3D.

These spatial object types and data types should comply with the definitions and constraints and include the attributes and association roles defined in this section.

The enumerations and code lists used in attributes or association roles of spatial object types or data types should comply with the definitions and include the values defined in this section.

5.2 Basic notions

This section explains some of the basic notions used in the INSPIRE application schemas. These explanations are based on the GCM [DS-D2.5].

5.2.1 Notation

5.2.1.1. Unified Modeling Language (UML)

The application schemas included in this section are specified in UML, version 2.1. The spatial object types, their properties and associated types are shown in UML class diagrams.

NOTE For an overview of the UML notation, see Annex D in [ISO 19103].

The use of a common conceptual schema language (i.e. UML) allows for an automated processing of application schemas and the encoding, querying and updating of data based on the application schema – across different themes and different levels of detail.

The following important rules related to class inheritance and abstract classes are included in the IRs.

IR Requirement

Article 5

Types

(...)

2. Types that are a sub-type of another type shall also include all this type's attributes and association roles.
3. Abstract types shall not be instantiated.

The use of UML conforms to ISO 19109 8.3 and ISO/TS 19103 with the exception that UML 2.1 instead of ISO/IEC 19501 is being used. The use of UML also conforms to ISO 19136 E.2.1.1.1-E.2.1.1.4.

NOTE ISO/TS 19103 and ISO 19109 specify a profile of UML to be used in conjunction with the ISO 19100 series. This includes in particular a list of stereotypes and basic types to be used in application schemas. ISO 19136 specifies a more restricted UML profile that allows for a direct encoding in XML Schema for data transfer purposes.

To model constraints on the spatial object types and their properties, in particular to express data/data set consistency rules, OCL (Object Constraint Language) is used as described in ISO/TS 19103, whenever possible. In addition, all constraints are described in the feature catalogue in English, too.

NOTE Since “void” is not a concept supported by OCL, OCL constraints cannot include expressions to test whether a value is a *void* value. Such constraints may only be expressed in natural language.

5.2.1.2. Stereotypes

In the application schemas in this section several stereotypes are used that have been defined as part of a UML profile for use in INSPIRE [DS-D2.5]. These are explained in Table 3 below.

Table 3 – Stereotypes (adapted from [DS-D2.5])

Stereotype	Model element	Description
applicationSchema	Package	An INSPIRE application schema according to ISO 19109 and the Generic Conceptual Model.
leaf	Package	A package that is not an application schema and contains no packages.
featureType	Class	A spatial object type.
type	Class	A type that is not directly instantiable, but is used as an abstract collection of operation, attribute and relation signatures. This stereotype should usually not be used in INSPIRE application schemas as these are on a different conceptual level than classifiers with this stereotype.
dataType	Class	A structured data type without identity.
union	Class	A structured data type without identity where exactly one of the properties of the type is present in any instance.
enumeration	Class	An enumeration.
codeList	Class	A code list.
import	Dependency	The model elements of the supplier package are imported.

voidable	Attribute, association role	A voidable attribute or association role (see section 5.2.2).
lifeCycleInfo	Attribute, association role	If in an application schema a property is considered to be part of the life-cycle information of a spatial object type, the property shall receive this stereotype.
version	Association role	If in an application schema an association role ends at a spatial object type, this stereotype denotes that the value of the property is meant to be a specific version of the spatial object, not the spatial object in general.

5.2.2 Voidable characteristics

The «voidable» stereotype is used to characterise those properties of a spatial object that may not be present in some spatial data sets, even though they may be present or applicable in the real world. This does *not* mean that it is optional to provide a value for those properties.

For all properties defined for a spatial object, a value has to be provided – either the corresponding value (if available in the data set maintained by the data provider) or the value of *void*. A *void* value shall imply that no corresponding value is contained in the source spatial data set maintained by the data provider or no corresponding value can be derived from existing values at reasonable costs.

Recommendation 2 The reason for a *void* value should be provided where possible using a listed value from the VoidReasonValue code list to indicate the reason for the missing value.

The VoidReasonValue type is a code list, which includes the following pre-defined values:

- *Unpopulated*: The property is not part of the dataset maintained by the data provider. However, the characteristic may exist in the real world. For example when the “elevation of the water body above the sea level” has not been included in a dataset containing lake spatial objects, then the reason for a void value of this property would be ‘Unpopulated’. The property receives this value for all spatial objects in the spatial data set.
- *Unknown*: The correct value for the specific spatial object is not known to, and not computable by the data provider. However, a correct value may exist. For example when the “elevation of the water body above the sea level” of a *certain lake* has not been measured, then the reason for a void value of this property would be ‘Unknown’. This value is applied only to those spatial objects where the property in question is not known.
- *Withheld*: The characteristic may exist, but is confidential and not divulged by the data provider.

NOTE It is possible that additional reasons will be identified in the future, in particular to support reasons / special values in coverage ranges.

The «voidable» stereotype does not give any information on whether or not a characteristic exists in the real world. This is expressed using the multiplicity:

- If a characteristic may or may not exist in the real world, its minimum cardinality shall be defined as 0. For example, if an Address may or may not have a house number, the multiplicity of the corresponding property shall be 0..1.
- If at least one value for a certain characteristic exists in the real world, the minimum cardinality shall be defined as 1. For example, if an Administrative Unit always has at least one name, the multiplicity of the corresponding property shall be 1..*.

In both cases, the «voidable» stereotype can be applied. In cases where the minimum multiplicity is 0, the absence of a value indicates that it is known that no value exists, whereas a value of void indicates that it is not known whether a value exists or not.

EXAMPLE If an address does not have a house number, the corresponding Address object should not have any value for the «voidable» attribute house number. If the house number is simply not

known or not populated in the data set, the Address object should receive a value of *void* (with the corresponding void reason) for the house number attribute.

5.2.3 Enumerations

Enumerations are modelled as classes in the application schemas. Their values are modelled as attributes of the enumeration class using the following modelling style:

- No initial value, but only the attribute name part, is used.
- The attribute name conforms to the rules for attributes names, i.e. is a lowerCamelCase name. Exceptions are words that consist of all uppercase letters (acronyms).

IR Requirement

Article 6

Code Lists and Enumerations

(...)

- 5) Attributes or association roles of spatial object types or data types that have an enumeration type may only take values from the lists specified for the enumeration type.”

5.2.4 Code lists

Code lists are modelled as classes in the application schemas. Their values, however, are managed outside of the application schema.

5.2.4.1. Code list types

The IRs distinguish the following types of code lists.

IR Requirement

Article 6

Code Lists and Enumerations

- 1) Code lists shall be of one of the following types, as specified in the Annexes:
 - a) code lists whose allowed values comprise only the values specified in this Regulation;
 - b) code lists whose allowed values comprise the values specified in this Regulation and narrower values defined by data providers;
 - c) code lists whose allowed values comprise the values specified in this Regulation and additional values at any level defined by data providers;
 - d) code lists, whose allowed values comprise any values defined by data providers.

For the purposes of points (b), (c) and (d), in addition to the allowed values, data providers may use the values specified in the relevant INSPIRE Technical Guidance document available on the INSPIRE web site of the Joint Research Centre.

The type of code list is represented in the UML model through the tagged value *extensibility*, which can take the following values:

- *none*, representing code lists whose allowed values comprise only the values specified in the IRs (type a);
- *narrower*, representing code lists whose allowed values comprise the values specified in the IRs and narrower values defined by data providers (type b);
- *open*, representing code lists whose allowed values comprise the values specified in the IRs and additional values at any level defined by data providers (type c); and

- *any*, representing code lists, for which the IRs do not specify any allowed values, i.e. whose allowed values comprise any values defined by data providers (type d).

Recommendation 3 Additional values defined by data providers should not replace or redefine any value already specified in the IRs.

NOTE This data specification may specify recommended values for some of the code lists of type (b), (c) and (d) (see section 5.2.4.3). These recommended values are specified in a dedicated Annex.

In addition, code lists can be hierarchical, as explained in Article 6(2) of the IRs.

IR Requirement

Article 6

Code Lists and Enumerations

(...)

- 2) Code lists may be hierarchical. Values of hierarchical code lists may have a more generic parent value. Where the valid values of a hierarchical code list are specified in a table in this Regulation, the parent values are listed in the last column.

The type of code list and whether it is hierarchical or not is also indicated in the feature catalogues.

5.2.4.2. Obligations on data providers

IR Requirement

Article 6

Code Lists and Enumerations

(....)

- 3) Where, for an attribute whose type is a code list as referred to in points (b), (c) or (d) of paragraph 1, a data provider provides a value that is not specified in this Regulation, that value and its definition shall be made available in a register.
- 4) Attributes or association roles of spatial object types or data types whose type is a code list may only take values that are allowed according to the specification of the code list.

Article 6(4) obliges data providers to use only values that are allowed according to the specification of the code list. The “allowed values according to the specification of the code list” are the values explicitly defined in the IRs plus (in the case of code lists of type (b), (c) and (d)) additional values defined by data providers.

For attributes whose type is a code list of type (b), (c) or (d) data providers may use additional values that are not defined in the IRs. Article 6(3) requires that such additional values and their definition be made available in a register. This enables users of the data to look up the meaning of the additional values used in a data set, and also facilitates the re-use of additional values by other data providers (potentially across Member States).

NOTE Guidelines for setting up registers for additional values and how to register additional values in these registers is still an open discussion point between Member States and the Commission.

5.2.4.3. Recommended code list values

For code lists of type (b), (c) and (d), this data specification may propose additional values as a recommendation (in a dedicated Annex). These values will be included in the INSPIRE code list

register. This will facilitate and encourage the usage of the recommended values by data providers since the obligation to make additional values defined by data providers available in a register (see section 5.2.4.2) is already met.

Recommendation 4 Where these Technical Guidelines recommend values for a code list in addition to those specified in the IRs, these values should be used.

NOTE For some code lists of type (d), no values may be specified in these Technical Guidelines. In these cases, any additional value defined by data providers may be used.

5.2.4.4. Governance

The following two types of code lists are distinguished in INSPIRE:

- *Code lists that are governed by INSPIRE (INSPIRE-governed code lists)*. These code lists will be managed centrally in the INSPIRE code list register. Change requests to these code lists (e.g. to add, deprecate or supersede values) are processed and decided upon using the INSPIRE code list register's maintenance workflows.

INSPIRE-governed code lists will be made available in the INSPIRE code list register at <http://inspire.ec.europa.eu/codelist/<CodeListName>>. They will be available in SKOS/RDF, XML and HTML. The maintenance will follow the procedures defined in ISO 19135. This means that the only allowed changes to a code list are the addition, deprecation or supersession of values, i.e. no value will ever be deleted, but only receive different statuses (valid, deprecated, superseded). Identifiers for values of INSPIRE-governed code lists are constructed using the pattern <http://inspire.ec.europa.eu/codelist/<CodeListName>/<value>>.

- *Code lists that are governed by an organisation outside of INSPIRE (externally governed code lists)*. These code lists are managed by an organisation outside of INSPIRE, e.g. the World Meteorological Organization (WMO) or the World Health Organization (WHO). Change requests to these code lists follow the maintenance workflows defined by the maintaining organisations. Note that in some cases, no such workflows may be formally defined.

Since the updates of externally governed code lists is outside the control of INSPIRE, the IRs and these Technical Guidelines reference a specific version for such code lists.

The tables describing externally governed code lists in this section contain the following columns:

- The *Governance* column describes the external organisation that is responsible for maintaining the code list.
- The *Source* column specifies a citation for the authoritative source for the values of the code list. For code lists, whose values are mandated in the IRs, this citation should include the version of the code list used in INSPIRE. The version can be specified using a version number or the publication date. For code list values recommended in these Technical Guidelines, the citation may refer to the "latest available version".
- In some cases, for INSPIRE only a subset of an externally governed code list is relevant. The subset is specified using the *Subset* column.
- The *Availability* column specifies from where (e.g. URL) the values of the externally governed code list are available, and in which formats. Formats can include machine-readable (e.g. SKOS/RDF, XML) or human-readable (e.g. HTML, PDF) ones.

Code list values are encoded using http URIs and labels. Rules for generating these URIs and labels are specified in a separate table.

Recommendation 5 The http URIs and labels used for encoding code list values should be taken from the INSPIRE code list registry for INSPIRE-governed code lists and generated according to the relevant rules specified for externally governed code lists.

NOTE Where practicable, the INSPIRE code list register could also provide http URIs and labels for externally governed code lists.

5.2.4.5. Vocabulary

For each code list, a tagged value called “vocabulary” is specified to define a URI identifying the values of the code list. For INSPIRE-governed code lists and externally governed code lists that do not have a persistent identifier, the URI is constructed following the pattern *http://inspire.ec.europa.eu/codelist/<UpperCamelCaseName>*.

If the value is missing or empty, this indicates an empty code list. If no sub-classes are defined for this empty code list, this means that any code list may be used that meets the given definition.

An empty code list may also be used as a super-class for a number of specific code lists whose values may be used to specify the attribute value. If the sub-classes specified in the model represent all valid extensions to the empty code list, the subtyping relationship is qualified with the standard UML constraint “{complete,disjoint}”.

5.2.5 Identifier management

IR Requirement

Article 9

Identifier Management

1. The data type Identifier defined in Section 2.1 of Annex I shall be used as a type for the external object identifier of a spatial object.
2. The external object identifier for the unique identification of spatial objects shall not be changed during the life-cycle of a spatial object.

NOTE 1 An external object identifier is a unique object identifier which is published by the responsible body, which may be used by external applications to reference the spatial object. [DS-D2.5]

NOTE 2 Article 9(1) is implemented in each application schema by including the attribute *inspireId* of type Identifier.

NOTE 3 Article 9(2) is ensured if the *namespace* and *localId* attributes of the Identifier remains the same for different versions of a spatial object; the *version* attribute can of course change.

5.2.6 Geometry representation

IR Requirement

Article 12

Other Requirements & Rules

1. The value domain of spatial properties defined in this Regulation shall be restricted to the Simple Feature spatial schema as defined in Herring, John R. (ed.), OpenGIS® Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture, version 1.2.1, Open Geospatial Consortium, 2011, unless specified otherwise for a specific spatial data theme or type.

NOTE 1 The specification restricts the spatial schema to 0-, 1-, 2-, and 2.5-dimensional geometries where all curve interpolations are linear and surface interpolations are performed by triangles.

NOTE 2 The topological relations of two spatial objects based on their specific geometry and topology properties can in principle be investigated by invoking the operations of the types defined in ISO 19107 (or the methods specified in EN ISO 19125-1).

5.2.7 Temporality representation

The application schema(s) use(s) the derived attributes "beginLifespanVersion" and "endLifespanVersion" to record the lifespan of a spatial object.

The attributes "beginLifespanVersion" specifies the date and time at which this version of the spatial object was inserted or changed in the spatial data set. The attribute "endLifespanVersion" specifies the date and time at which this version of the spatial object was superseded or retired in the spatial data set.

NOTE 1 The attributes specify the beginning of the lifespan of the version in the spatial data set itself, which is different from the temporal characteristics of the real-world phenomenon described by the spatial object. This lifespan information, if available, supports mainly two requirements: First, knowledge about the spatial data set content at a specific time; second, knowledge about changes to a data set in a specific time frame. The lifespan information should be as detailed as in the data set (i.e., if the lifespan information in the data set includes seconds, the seconds should be represented in data published in INSPIRE) and include time zone information.

NOTE 2 Changes to the attribute "endLifespanVersion" does not trigger a change in the attribute "beginLifespanVersion".

IR Requirement

Article 10

Life-cycle of Spatial Objects

(...)

3. Where the attributes beginLifespanVersion and endLifespanVersion are used, the value of endLifespanVersion shall not be before the value of beginLifespanVersion.

NOTE The requirement expressed in the IR Requirement above will be included as constraints in the UML data models of all themes.

Recommendation 6

If life-cycle information is not maintained as part of the spatial data set, all spatial objects belonging to this data set should provide a void value with a reason of "unpopulated".

5.3 Application schema BuildingsBase

5.3.1 Description

5.3.1.1. Narrative description

Buildings Base application schema is an abstract application schema that describes the feature types, data types and code lists that are common to all the four instanciable application schemas, namely 2D, 3D, extended2D and extended3D.

It addresses mainly the basic normative semantics and includes in addition a data type about the 2D geometric representation of buildings that is used by all the four instanciable application schemas.

5.3.1.1.1. Feature types

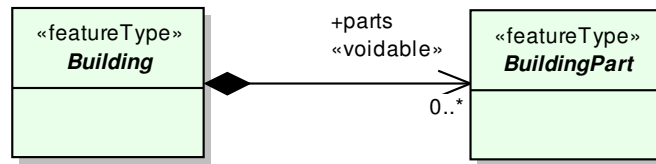


Figure 12: Instanciable feature types

Buildings are enclosed constructions above and/or underground which are intended or used for the shelter of humans, animals, things or the production of economic goods and that refer to any structure permanently constructed or erected on its site.

According to a CityGML concept, a complex building may be considered as an aggregation of BuildingParts, as shown on the following illustration:



A BuildingPart is a sub-division of a Building that might have been considered as a building and that is homogeneous related to its physical, functional or temporal aspects. It is up to each data producer to define what is considered as a Building and what is considered as a BuildingPart (if this concept is used). This information has to be provided as metadata.

More explanations and examples about how the concept of BuildingPart may and should be used are provided in clause 10 about Data capture.

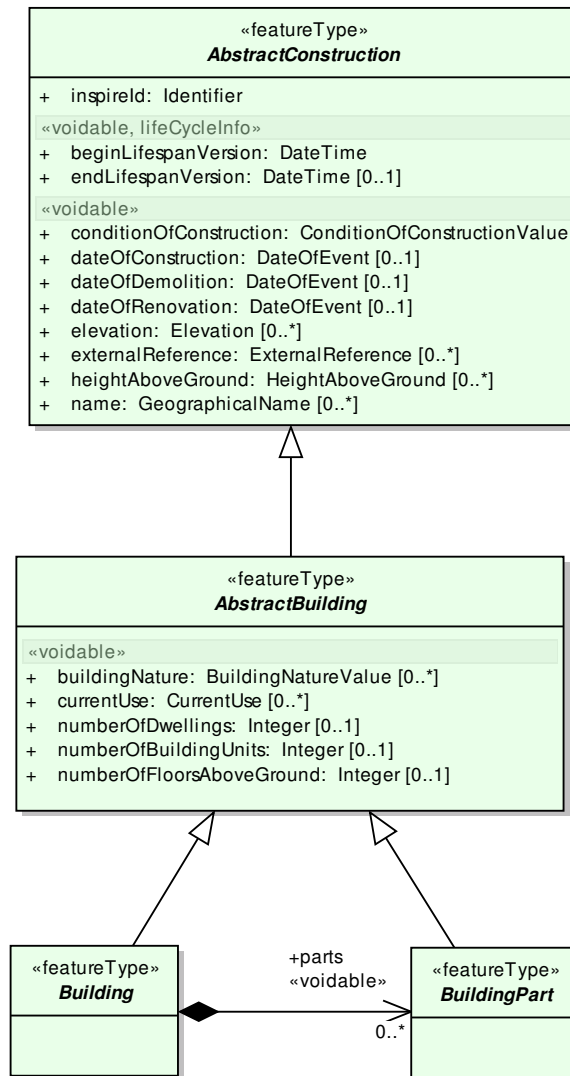


Figure 13: Feature types of Buildings Base application schema

Base application schema includes 2 abstract feature types: **AbstractConstruction** and **AbstractBuilding**:

- **AbstractBuilding** is an abstract feature type grouping the common properties of instanciable feature types **Building** and **BuildingPart**
- **AbstractConstruction** is an abstract feature type grouping the semantic properties of buildings, building parts and of some optional feature types that may be added to core profiles, in order to provide more information about theme *Buildings*. The optional feature types are described in extended application schemas.

Instanciable feature types **Building** and **BuildingPart** inherit both of the properties of abstract feature types **AbstractConstruction** and **AbstractBuilding**.

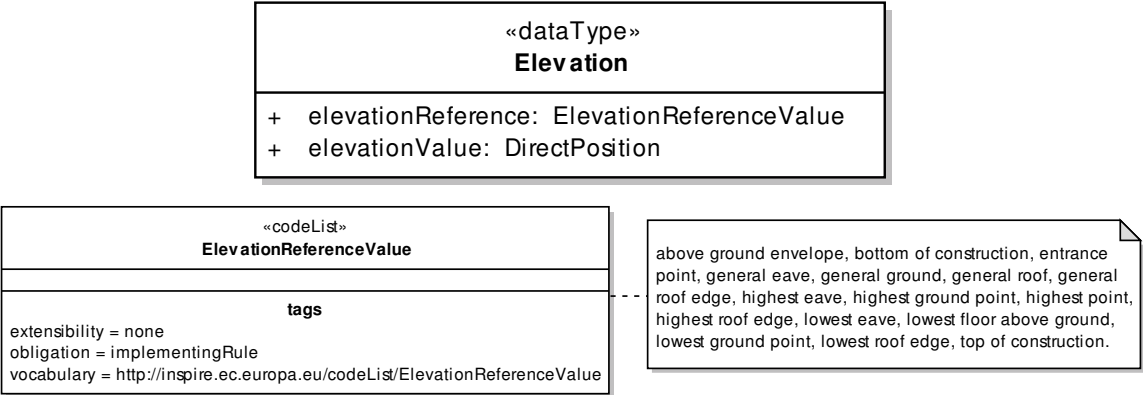


Figure 14: The Elevation data type

A building or a construction may have several values of attribute elevation:

- the elevation may be measured at different levels of the building; this must be documented with attribute elevationReference, using the possible values given in the code list ElevationReferenceValue (see Figure 15: Examples of elevation references for different kinds of building)
- the elevation may be given in various vertical reference systems; this has to be documented through the DirectPosition that contains both the elevation value itself and the Coordinate Reference System to which this value refers.

EXAMPLE (DirectPosition):
<pos srsName="urn:x-ogc:def:crs:EPSG:7.9.5. 5621" srsDimension="1">114</pos>
The Spatial Reference System Name (srsName) is given by:

- EPSG :7.9.5 : namespace (or register) and its version
- 5621 : identifier of the CRS in the given register (here it is EVRF 2007)

The srsDimension is 1 because related only to one dimension (elevation).
The value of elevation is 114.

Recommendation 7 For territories that are in the scope of EVRS, the use EVRS as elevation datum is recommended.

However, some communities as marine or air navigation may have other requirements, coming from international standards.

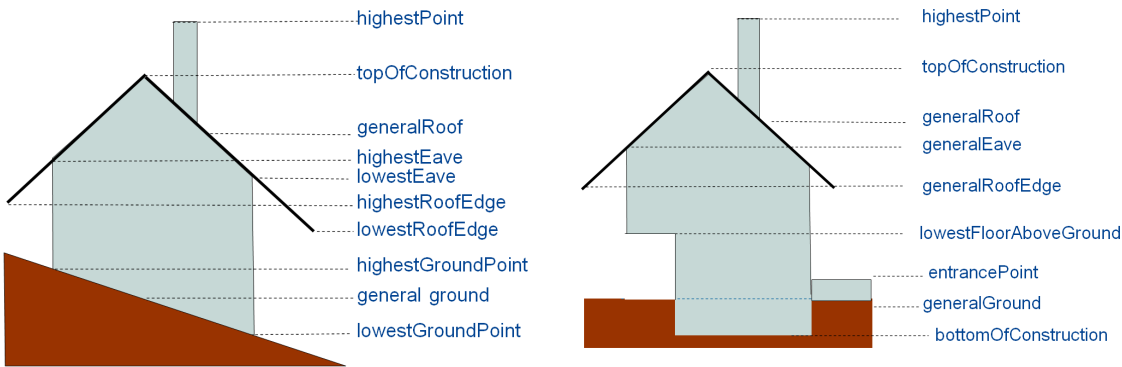


Figure 15: Examples of elevation references for different kinds of building

5.3.1.1.3. *Attribute HeightAboveGround*

A construction of a building may have several values for the attribute HeightAboveGround, according to the levels that were chosen to compute it. The heightAboveGround of a construction or building is generally computed as the difference between an elevation measured at a high reference and the elevation measured at a low reference.

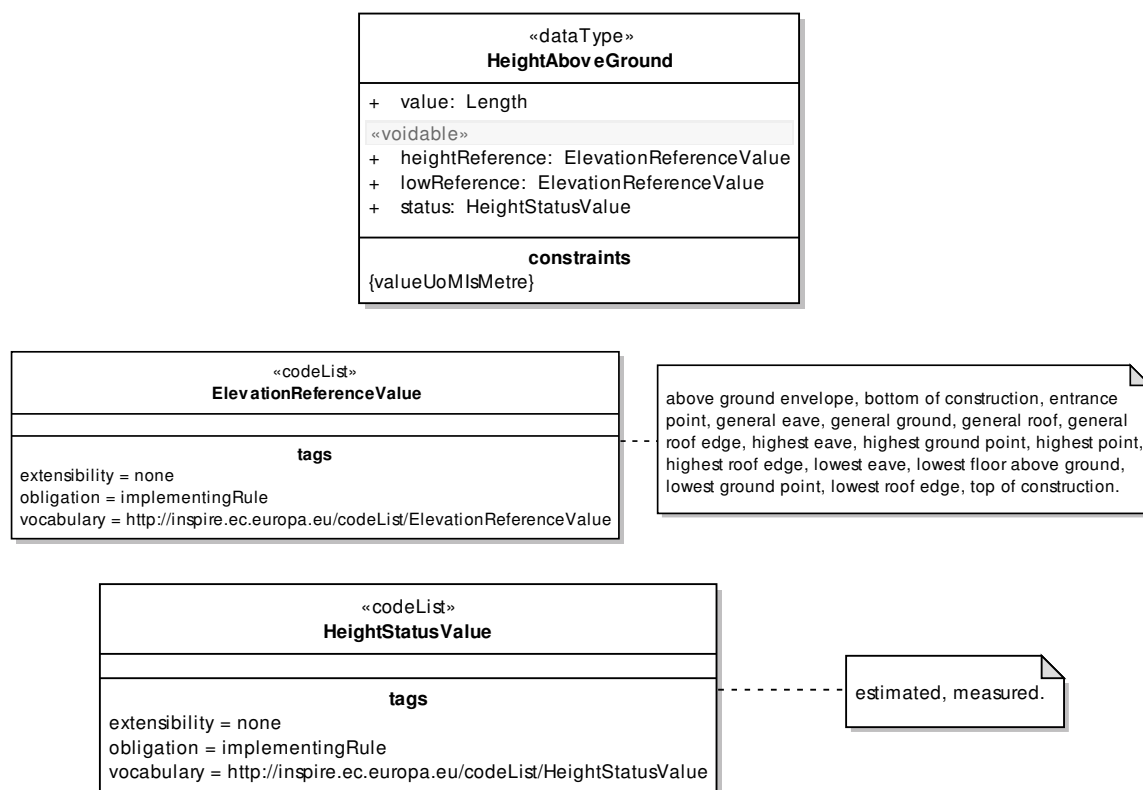


Figure 16: The HeightAboveGround data type

It is recommended to use:

- For the low reference
 - generalGround
 - lowestGroundPoint
 - lowestFloorAboveGround
 - entrancePoint
 - highestGroundPoint
- For the high reference
 - generalRoofEdge
 - lowestRoofEdge
 - highestRoofEdge
 - generalEave
 - lowestEave
 - highestEave
 - generalRoof
 - top OfConstruction
 - highestPoint

5.3.1.1.4. Classification of buildings

The classification of buildings has to be done using two attributes:

- the attribute `currentUse` that focuses on the activity hosted by the building; this attribute aims to fulfil management requirements, such as computation of population or spatial planning ; this classification aims to be exhaustive for the functional buildings hosting human activities
- the attribute `buildingNature` that focuses on the physical aspect of the building; however, this physical aspect is often expressed as a function (e.g. stadium, silo, windmill); this attribute aims to fulfil mainly mapping purposes and addresses only specific, noticeable buildings. This is a rather short and simple list of possible values, with focus on two international use cases: air flights where buildings may be obstacles and marine navigation where buildings may be landmarks.

The code list for attribute `buildingNature` may be extended by Member States, in order to fulfil more mapping requirements.

The attribute `currentUse` may take its possible values in a hierarchical code list. This hierarchical code list should enable easy matching from existing classifications to the INSPIRE classification:

- a data producer with simple classification may match at the upper level of INSPIRE classification (e.g. residential / agriculture / industrial / commerceAndService)
- a data producer with a more detailed classification may match at the lower levels of INSPIRE classification (e.g. `moreThanTwoDwellings`, `publicServices`, ...).

The code list for attribute `currentUse` may also be extended by Member States, but only by providing more detailed values, under the hierarchical structure of the INSPIRE code list.

Some examples are provided in the annex F.

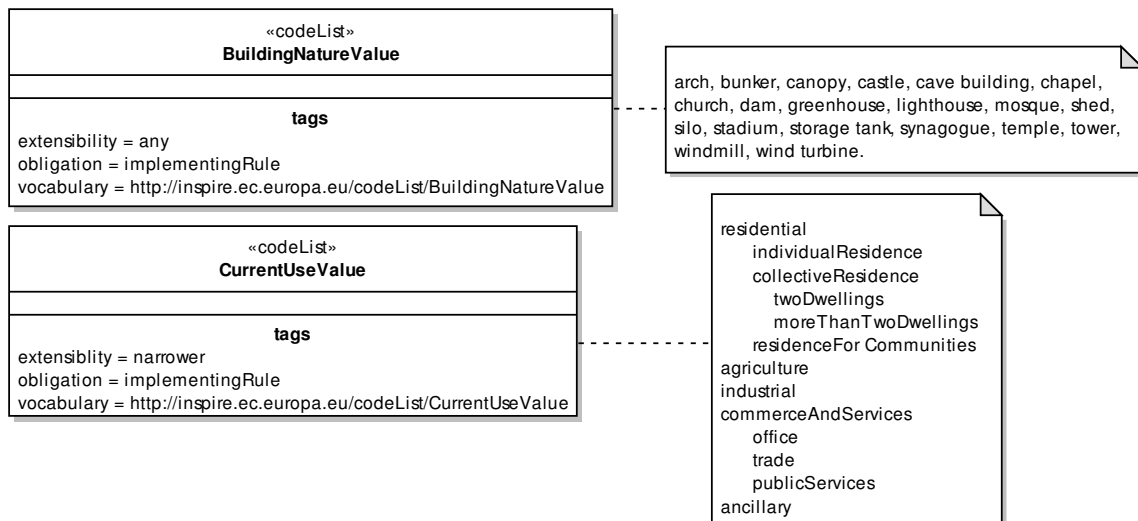
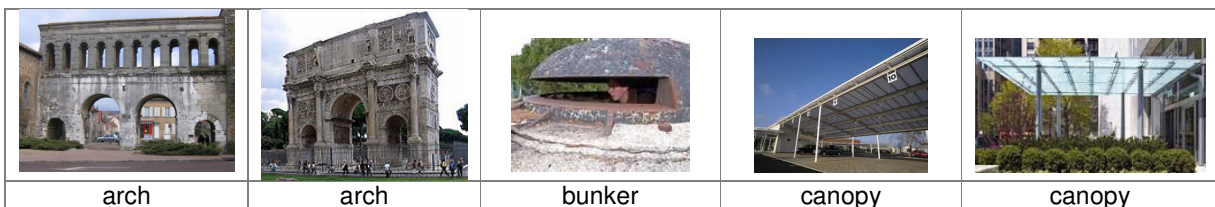


Figure 17: Code lists for classification of buildings




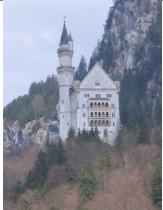





















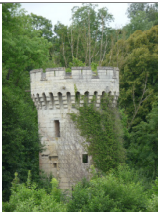






				
castle	castle	caveBuilding	caveBuilding	chapel
				
Chapel	church	church	dam	dam
				
greenhouse	greenhouse	lighthouse	mosque	shed
				
shed	silo	stadium	stadium	storageTank
				
synagogue	temple	temple	tower	tower
				
tower	tower	transformer	windmill	windTurbine

Figure 18: Building nature

5.3.1.1.5. *Attribute externalReference*

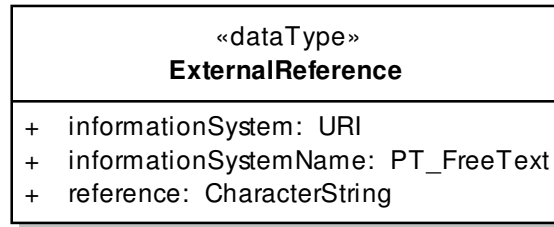


Figure 19: The attribute externalReference is defined as a data type

This attribute aims to ensure the link to other information systems, for instance:

- another spatial data set including building data; in this case, the external reference contributes to ensure consistency between different views or different levels of detail on same real-world objects, that is an explicit requirement of the INSPIRE Directive,
- the cadastral register where information about owner, tenant, criteria of valuation (heating, toilet, ...) may be found.

5.3.1.1.6. *Geometry of buildings*

All instanciable application schema include an attribute geometry2D, with multiplicity [1..*]. This attribute is mandatory in 2D profiles and voidable in 3D profiles. The *Buildings* base application schema does not contain the attribute geometry2D itself but it describes the data type used to represent it: BuildingGeometry2D.

The INSPIRE model is quite flexible as it allows the geometry of a building to be represented in different ways. Multiple geometries are allowed for buildings; for instance, a data producer may provide representation of a building as a surface and as a point or as several surfaces, e.g. the building captured by its foot print and by its roof edges.

Whereas the representation by surfaces is expected by most use cases, the representation by point is useful to make some computations quicker (e.g. computation of distances).

However, a view service may only use one geometry; the geometry to be chosen by the view service is documented through the Boolean attribute **referenceGeometry**: there shall be only one geometric representation whose attribute referenceGeometry gets the value "true". In case of representation by point and by surface, the surface should generally be the reference geometry.

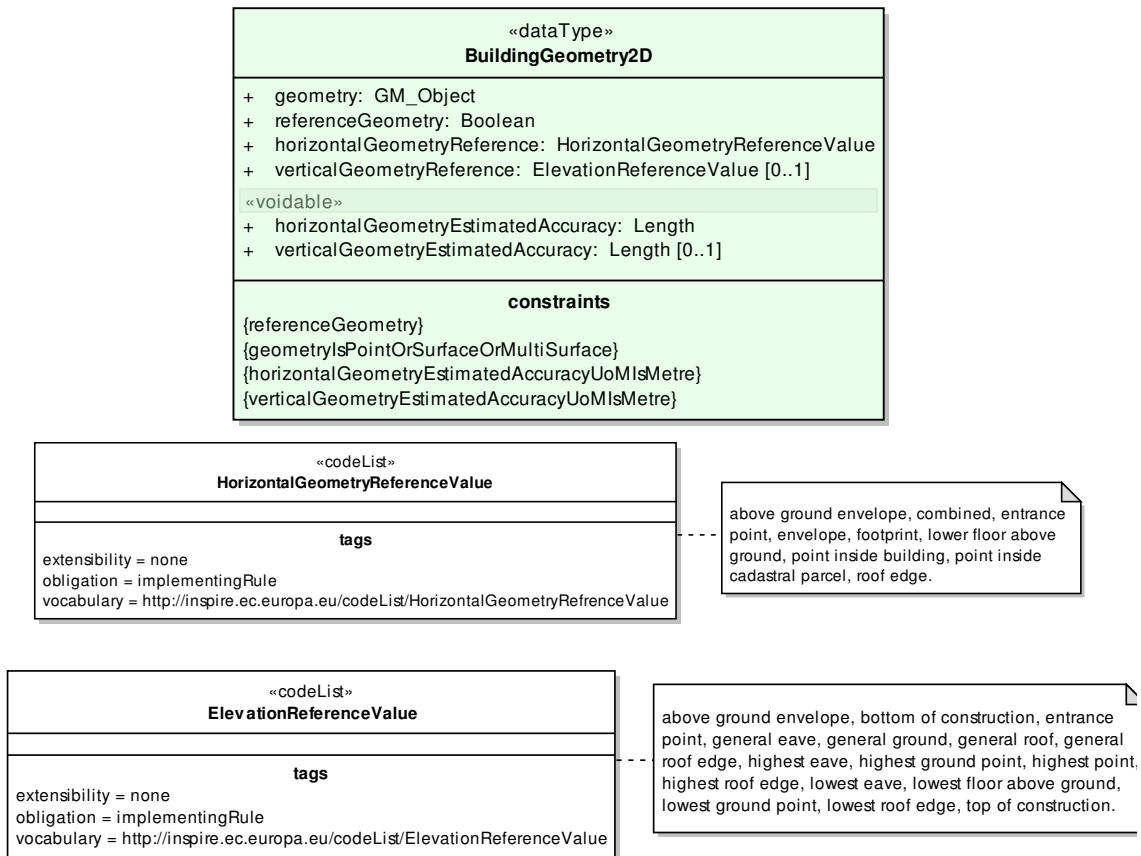
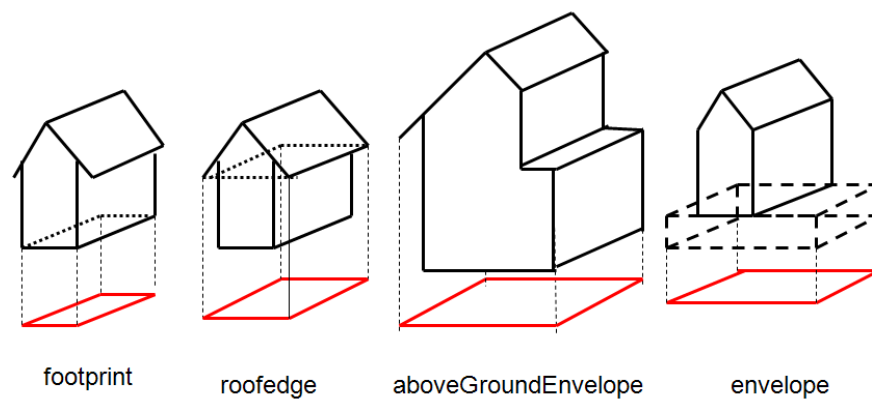


Figure 20: The geometry of Building and BuildingPart has to be documented

A building is a 3D object represented in this profile by 2D or 2,5 D data:

- the place where (X,Y) coordinates were captured has to be documented using the attribute horizontalGeometryReference;
- the place where Z coordinate was captured must be documented using the attribute verticalGeometryReference.



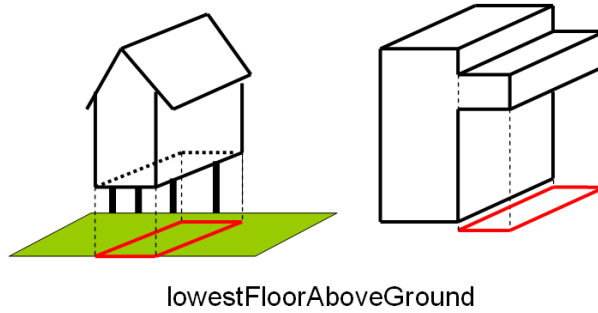


Figure 21: Examples of HorizontalGeometryReference

NOTE: The possible values of attribute horizontalGeometryReference depend on the geometric representation of the building or building part, as shown in the Table 4 below.

Table 4: Correspondence between geometry and horizontalGeometryReference

geometry	GM_Point	GM_Surface GM_MultiSurface
horizontalGeometryReference	entrancePoint pointInsideBuilding pointInsideCadastralParcel	Footprint Roofedge aboveGroundEnvelope envelope lowestFloorAboveGround combined

NOTE: it is not forbidden to represent different levels of detail of the same building. The model allows for instance to represent the geometries of the building, captured at different scales, using the same horizontal geometry reference., e.g. a building captured by its roof edge with different generalisation rules or from aerial images taken at different original scales. In this case, it is strongly recommended to provide the attribute horizontalGeometryEstimatedAccuracy and/or to give referenceGeometry to the most detailed one.

5.3.1.2. UML Overview

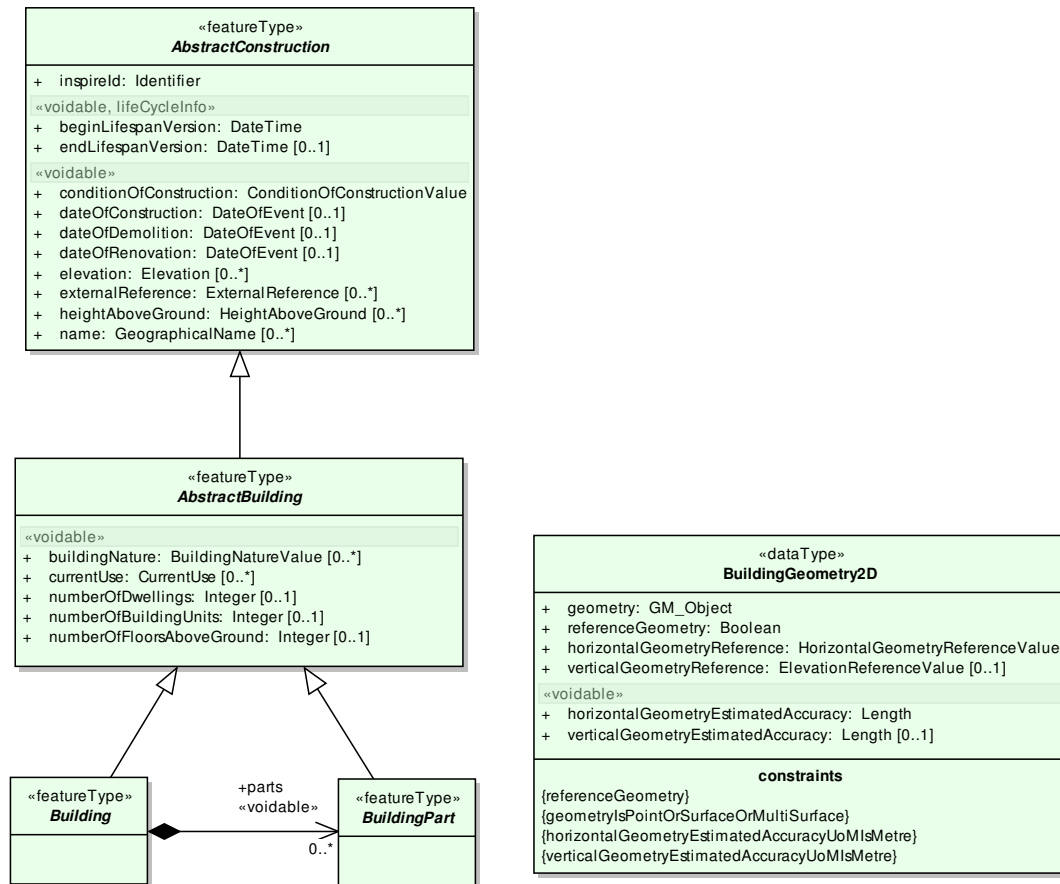


Figure 22: UML class diagram: Overview of the Building Base - Main types

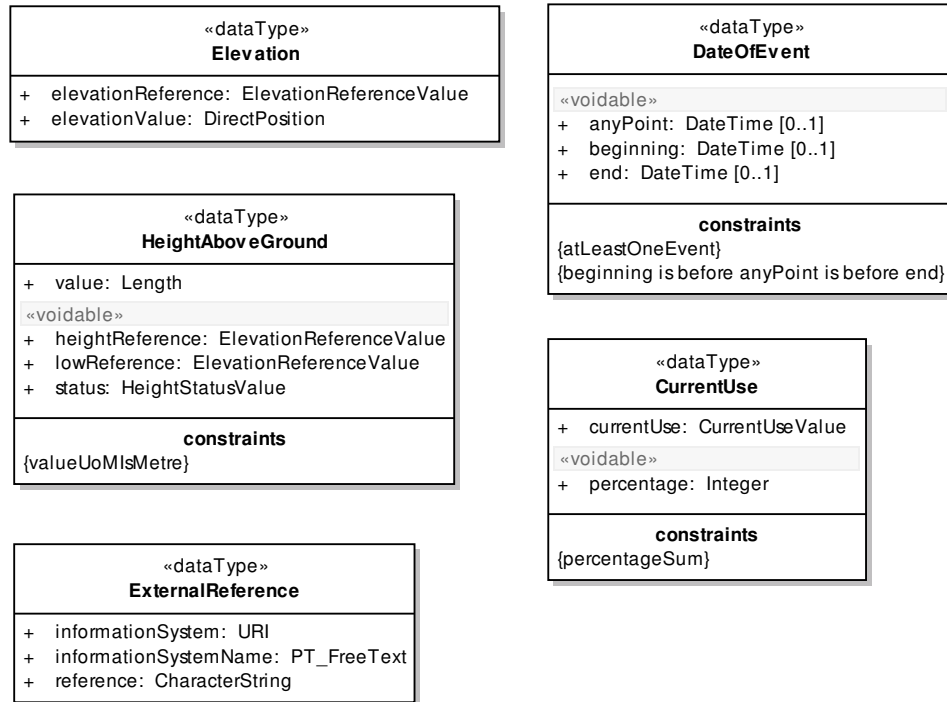


Figure 23: UML class diagram: Overview of the Building Base - Data types

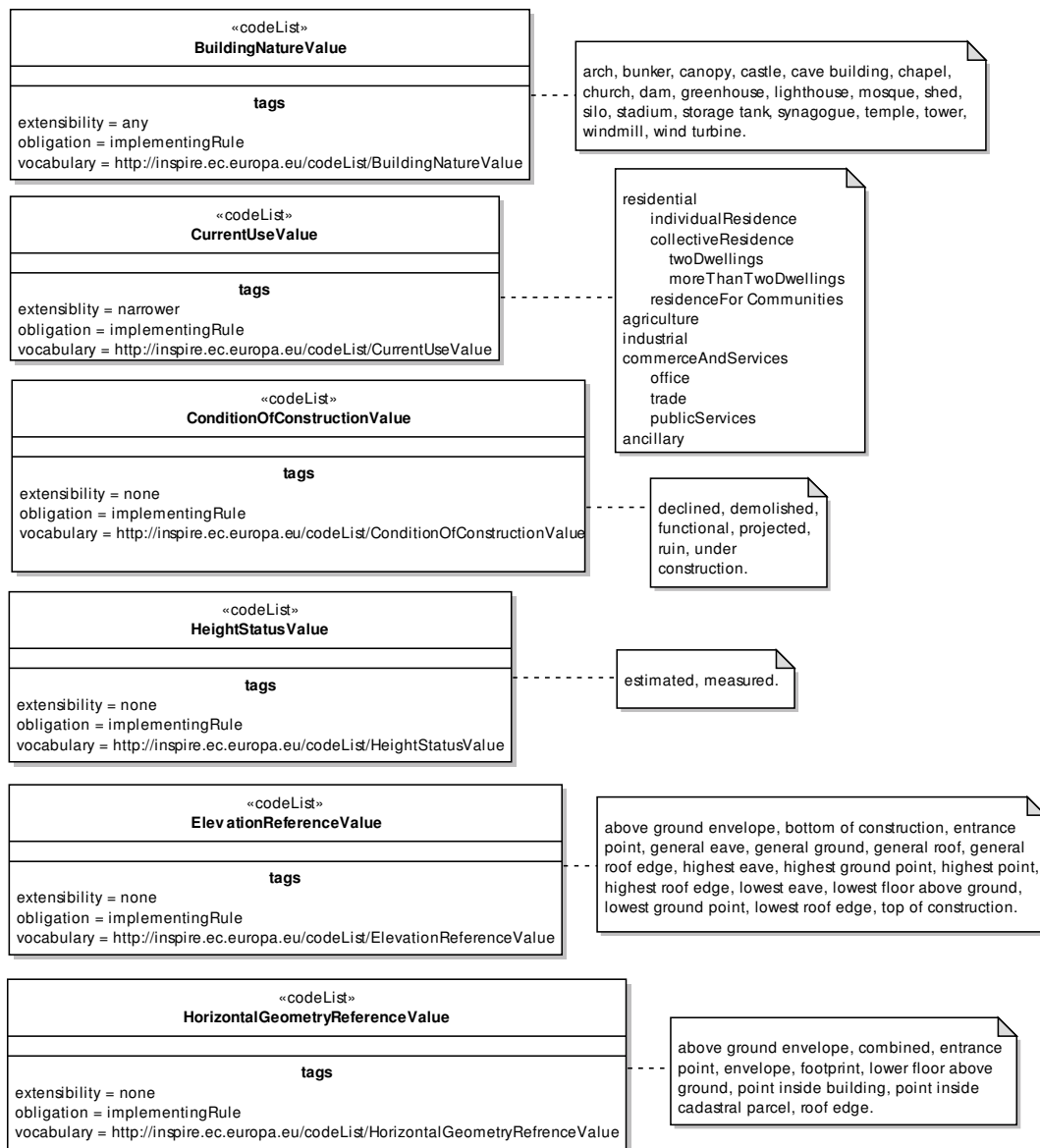


Figure 24: UML class diagram: Overview of the Building Base - Code lists

5.3.1.3. Consistency between spatial data sets

There should be some consistency between the value of attribute *currentUse* in theme *Buildings* and the location of agricultural facilities, industrial or production facilities and governmental services. Typically:

- *Buildings* within an agricultural or aquacultural facility should generally have value “agricultural” for attribute *current use*
- *Buildings* within an industrial or production facility should generally have value “industrial” for attribute *current use*
- *Buildings* within a governmental service should generally have value “public services” for attribute *current use*

However, there may be exceptions (e.g. a residential building for guardian in a production site or for teacher in a school or for farmer in an agricultural facility); moreover, the geometry of facilities or governmental services may be represented in some cases just by a point and so, may not always enable to identified the related buildings. Consequently, no absolute consistency rule can be provided.

Recommendation 8	Member States and/or National Spatial Data Infrastructures should encourage cooperation between data providers of themes <i>Buildings</i> and of themes Agricultural and Aquacultural Facilities, Production and Industrial facilities and Utility and Governmental Services in order to provide consistent data.
-------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.3.1.4. Identifier management

The buildings and building parts have to be identified by the mandatory attribute inspireID; this unique identification enables the buildings and building parts to be target of associations from other INSPIRE themes, e.g. from theme Address.

5.3.1.5. Modelling of object references

The base application schema offers one option to link a spatial object (building or building part) defined in INSPIRE to information in other systems: the attribute externalReference provides the identifier/reference of the object in that foreign system together with the name and the URI of that information system. This external reference for instance may be used to obtain information about the owner of the building from a cadastral system.

The external information systems that may/should be linked to theme *Buildings* depend of course of national context and regulations.

Recommendation 9	Member States and/or National Spatial Data Infrastructures should agree on the external information systems to be linked to theme <i>Buildings</i> .
-------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------

5.3.2 Feature catalogue

Feature catalogue metadata

Application Schema	INSPIRE Application Schema BuildingsBase
Version number	3.0

Types defined in the feature catalogue

Type	Package	Stereotypes
<i>AbstractBuilding</i>	BuildingsBase	«featureType»
<i>AbstractConstruction</i>	BuildingsBase	«featureType»
<i>Building</i>	BuildingsBase	«featureType»
<i>BuildingGeometry2D</i>	BuildingsBase	«dataType»
<i>BuildingNatureValue</i>	BuildingsBase	«codeList»
<i>BuildingPart</i>	BuildingsBase	«featureType»
<i>ConditionOfConstructionValue</i>	BuildingsBase	«codeList»
<i>CurrentUse</i>	BuildingsBase	«dataType»
<i>CurrentUseValue</i>	BuildingsBase	«codeList»
<i>DateOfEvent</i>	BuildingsBase	«dataType»
<i>Elevation</i>	BuildingsBase	«dataType»
<i>ElevationReferenceValue</i>	BuildingsBase	«codeList»
<i>ExternalReference</i>	BuildingsBase	«dataType»
<i>HeightAboveGround</i>	BuildingsBase	«dataType»
<i>HeightStatusValue</i>	BuildingsBase	«codeList»
<i>HorizontalGeometryReferenceValue</i>	BuildingsBase	«codeList»

5.3.2.1. Spatial object types

5.3.2.1.1. AbstractConstruction

AbstractConstruction (abstract)	
Name:	Abstract construction
Definition:	Abstract spatial object type grouping the semantic properties of buildings, building parts and of some optional spatial object types that may be added in order to provide more information about the theme <i>Buildings</i> .
Description:	The optional spatial object types that may be added to core profiles are described in the extended profiles. The ones inheriting from the attributes of AbstractConstruction are Installation and OtherConstruction.
Stereotypes:	«featureType»
Attribute: beginLifespanVersion	
Name:	Begin lifespan version
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
Multiplicity:	1
Stereotypes:	«voidable,lifeCycleInfo»
Attribute: conditionOfConstruction	
Name:	Condition of construction
Value type:	ConditionOfConstructionValue
Definition:	Status of the construction.
Description:	EXAMPLES: functional, projected, ruin
Multiplicity:	1

AbstractConstruction (abstract)	
Stereotypes:	«voidable»
Attribute: dateOfConstruction	
Name:	Date of construction
Value type:	DateOfEvent
Definition:	Date of construction.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: dateOfDemolition	
Name:	Date of demolition
Value type:	DateOfEvent
Definition:	Date of demolition.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: dateOfRenovation	
Name:	Date of last major renovation
Value type:	DateOfEvent
Definition:	Date of last major renovation.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: elevation	
Name:	Elevation
Value type:	Elevation
Definition:	Vertically-constrained dimensional property consisting of an absolute measure referenced to a well-defined surface which is commonly taken as origin (geoïd, water level, etc.).
Description:	Source: adapted from the definition given in the data specification of the theme Elevation.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: endLifespanVersion	
Name:	End lifespan version
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
Multiplicity:	0..1
Stereotypes:	«voidable,lifeCycleInfo»
Attribute: externalReference	
Name:	External reference
Value type:	ExternalReference
Definition:	Reference to an external information system containing any piece of information related to the spatial object.
Description:	<p>EXAMPLE 1: Reference to another spatial data set containing another view on buildings; the externalReference may be used for instance to ensure consistency between 2D and 3D representations of the same buildings</p> <p>EXAMPLE 2: Reference to cadastral or dwelling register. The reference to this register may enable to find legal information related to the building, such as the owner(s) or valuation criteria (e.g. type of heating, toilet, kitchen)</p>

AbstractConstruction (abstract)	
	EXAMPLE 3: Reference to the system recording the building permits. The reference to the building permits may be used to find detailed information about the building physical and temporal aspects.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: heightAboveGround	
Name:	Height above ground
Value type:	HeightAboveGround
Definition:	Height above ground.
Description:	NOTE: height above ground may be defined as the difference between elevation at a low reference (ground level) and elevation as a high reference (e.g. roof level, top of construction)
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: inspireId	
Name:	inspire id
Value type:	Identifier
Definition:	External object identifier of the spatial object.
Description:	An external object identifier is a unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object. The identifier is an identifier of the spatial object, not an identifier of the real-world phenomenon.
Multiplicity:	1
Attribute: name	
Value type:	GeographicalName
Definition:	Name of the construction.
Description:	EXAMPLES: Big Ben, Eiffel Tower, Sacrada Familia
Multiplicity:	0..*
Stereotypes:	«voidable»

5.3.2.1.2. *AbstractBuilding*

AbstractBuilding (abstract)	
Name:	Abstract building
Subtype of:	AbstractConstruction
Definition:	Abstract spatial object type grouping the common semantic properties of the spatial object types Building and BuildingPart.
Stereotypes:	«featureType»
Attribute: buildingNature	
Name:	Building nature
Value type:	BuildingNatureValue
Definition:	Characteristic of the building that makes it generally of interest for mappings applications. The characteristic may be related to the physical aspect and/or to the function of the building.
Description:	This attribute focuses on the physical aspect of the building; however, this physical aspect is often expressed as a function (e.g. stadium, silo, windmill); this attribute aims to fulfil mainly mapping purposes and addresses only specific, noticeable buildings.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: currentUse	

AbstractBuilding (abstract)	
Name:	Current use
Value type:	CurrentUse
Definition:	Activity hosted within the building. This attribute addresses mainly the buildings hosting human activities.
Description:	NOTE: . This attribute aims to fulfill management requirements, such as computation of population or spatial planning ; this classification aims to be exhaustive for the functional buildings hosting human activities.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: numberOfDwellings	
Name:	Number of dwellings
Value type:	Integer
Definition:	Number of dwellings.
Description:	A dwelling is a residential unit which may consist of one or several rooms designed for the occupation of households. NOTE: In the data sets including building units, a dwelling is a residential building unit or, only when that building has no building units, a residential building. EXAMPLES: a single building dwelling could be a detached or semi-detached house. A block of flats will contain multiple dwellings determined by the number of individual flats.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: numberOfBuildingUnits	
Name:	Number of building units
Value type:	Integer
Definition:	Number of building units in the building. A BuildingUnit is a subdivision of Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc.
Description:	Building units are spatial objects aimed at subdividing buildings and/or building parts into smaller parts that are treated as separate entities in daily life. A building unit is homogeneous, regarding management aspects. EXAMPLES: It may be e.g. an apartment in a condominium, a terraced house, or a shop inside a shopping arcade. NOTE 1: According to national regulations, a building unit may be a flat, a cellar, a garage or set of a flat, a cellar and a garage. NOTE 2: According to national regulation, a building that is one entity for daily life (typically, a single family house) may be considered as a Building composed of one BuildingUnit or as a Building composed of zero BuildingUnit.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: numberOfFloorsAboveGround	
Name:	Number of floors above ground
Value type:	Integer
Definition:	Number of floors above ground.
Multiplicity:	0..1
Stereotypes:	«voidable»
5.3.2.1.3. Building	
Building (abstract)	
Name:	Building

Building (abstract)	
Subtype of:	AbstractBuilding
Definition:	A Building is an enclosed construction above and/or underground, used or intended for the shelter of humans, animals or things or for the production of economic goods. A building refers to any structure permanently constructed or erected on its site.
Stereotypes:	«featureType»
Association role: parts	
Value type:	BuildingPart
Definition:	The building parts composing the Building.
Description:	A building may be a simple building (with no BuildingPart) or a composed building (with several BuildingParts).
Multiplicity:	0..*
Stereotypes:	«voidable»

5.3.2.1.4. *BuildingPart*

BuildingPart (abstract)	
Name:	Building part
Subtype of:	AbstractBuilding
Definition:	A BuildingPart is a sub-division of a Building that might be considered itself as a building.
Description:	NOTE 1: A BuildingPart is homogeneous related to its physical, functional or temporal aspects. NOTE 2: Building and BuildingPart share the same set of properties. EXAMPLE: A building may be composed of two building parts having different heights above ground.
Stereotypes:	«featureType»

5.3.2.2. Data types

5.3.2.2.1. *DateOfEvent*

DateOfEvent	
Name:	Date of event
Definition:	This data type includes the different possible ways to define the date of an event.
Stereotypes:	«dataType»
Attribute: anyPoint	
Name:	Any point
Value type:	DateTime
Definition:	A date and time of any point of the event, between its beginning and its end.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: beginning	
Name:	Beginning
Value type:	DateTime
Definition:	Date and time when the event begun.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: end	
Name:	End
Value type:	DateTime
Definition:	Date and time when the event ended.

DateOfEvent	
Multiplicity:	0..1
Stereotypes:	«voidable»
Constraint: atLeastOneEvent	
Natural language:	At least, one of the attributes beginning, end or anyPoint shall be supplied.
OCL:	inv: dateOfEvent->notEmpty()
Constraint: beginning is before anyPoint is before end	
Natural language:	
OCL:	inv: beginning <= anyPoint and anyPoint <= end and beginning <= end

5.3.2.2.2. HeightAboveGround

HeightAboveGround	
Name:	Height above ground
Definition:	Vertical distance (measured or estimated) between a low reference and a high reference.
Stereotypes:	«dataType»
Attribute: heightReference	
Name:	Height reference
Value type:	ElevationReferenceValue
Definition:	Element used as the high reference.
Description:	EXAMPLE: The height of the building has been captured up to the top of building.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: lowReference	
Name:	Low reference
Value type:	ElevationReferenceValue
Definition:	Element as the low reference.
Description:	EXAMPLE: the height of the building has been captured from its the lowest ground point.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: status	
Name:	Status
Value type:	HeightStatusValue
Definition:	The way the height has been captured.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: value	
Name:	Value
Value type:	Length
Definition:	Value of the height above ground.
Multiplicity:	1
Constraint: valueUoMIsMetre	
Natural language:	Value shall be in meters.
OCL:	inv: self.value.uom.uomSymbol='m'

5.3.2.2.3. Elevation

Elevation	
Name:	Elevation
Definition:	This data types includes the elevation value itself and information on how this elevation was measured.
Stereotypes:	«dataType»
Attribute: elevationReference	
Name:	Elevation reference
Value type:	ElevationReferenceValue
Definition:	Element where the elevation was measured.
Multiplicity:	1
Attribute: elevationValue	
Name:	elevation value
Value type:	DirectPosition
Definition:	Value of the elevation.
Multiplicity:	1

5.3.2.2.4. BuildingGeometry2D

BuildingGeometry2D	
Name:	Building geometry 2D
Definition:	This data types includes the geometry of the building and metadata information about which element of the building was captured and how.
Stereotypes:	«dataType»
Attribute: geometry	
Name:	Geometry
Value type:	GM_Object
Definition:	2D or 2.5D geometric representation
Multiplicity:	1
Attribute: referenceGeometry	
Name:	Reference geometry
Value type:	Boolean
Definition:	The geometry to be taken into account by view services, for portrayal.
Description:	NOTE 1: In case of multiple representation by point and by surface, it is generally recommended to provide the surface as reference geometry. NOTE 2: The geometric representation whose referenceGeometry is true may also be used preferably for spatial queries by download services (WFS) or by Geographical Information System (GIS).
Multiplicity:	1
Attribute: horizontalGeometryReference	
Name:	Horizontal geometry reference
Value type:	HorizontalGeometryReferenceValue
Definition:	Element of the building that was captured by (X,Y) coordinates.
Multiplicity:	1
Attribute: verticalGeometryReference	
Name:	Vertical geometry reference
Value type:	ElevationReferenceValue
Definition:	Element of the building that was captured by vertical coordinates.
Multiplicity:	0..1
Attribute: horizontalGeometryEstimatedAccuracy	

BuildingGeometry2D	
Name:	Horizontal geometry estimated accuracy
Value type:	Length
Definition:	The estimated absolute positional accuracy of the (X,Y) coordinates of the building geometry, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE: This mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: verticalGeometryEstimatedAccuracy	
Name:	Vertical geometry estimated accuracy
Value type:	Length
Definition:	The estimated absolute positional accuracy of the Z coordinates of the building geometry, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE: This mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Constraint: geometryIsPointOrSurfaceOrMultiSurface	
Natural language:	Geometry shall be of type GM_Point or GM_Surface or GM_MultiSurface.
OCL:	
Constraint: horizontalGeometryEstimatedAccuracyUoMIsMetre	
Natural language:	The value of horizontalGeometryEstimatedAccuracy shall be given in meters.
OCL:	inv: self.horizontalGeometryEstimatedAccuracy.uom.uomSymbol='m'
Constraint: referenceGeometry	
Natural language:	For exactly one item of BuildingGeometry, the value of the attribute referenceGeometry shall be 'true'.
OCL:	
Constraint: verticalGeometryEstimatedAccuracyUoMIsMetre	
Natural language:	The Value of verticalGeometryEstimatedAccuracy has to be given in meters.
OCL:	inv: self.verticalGeometryEstimatedAccuracy.uom.uomSymbol='m'
5.3.2.2.5. <i>CurrentUse</i>	
CurrentUse	
Name:	Current use
Definition:	This data type enables to detail the current use(s).
Stereotypes:	«dataType»
Attribute: currentUse	

CurrentUse	
Name:	Current use
Value type:	CurrentUseValue
Definition:	The current use.
Description:	EXAMPLE: trade
Multiplicity:	1
Attribute: percentage	
Name:	Percentage
Value type:	Integer
Definition:	The proportion of the real world object, given as a percentage, devoted to this current use.
Description:	NOTE: The percentage of use is generally the percentage of floor area dedicated to this given use. If it is not the case, it is recommended to explain what the percentage refers to in metadata (template for additional information) EXAMPLE: 30 (if 30% of the building is occupied by trade activity).
Multiplicity:	1
Stereotypes:	«voidable»
Constraint: percentageSum	
Natural language:	The total of all percentages shall be less or equal to 100.
OCL:	inv: self.percentage.sum()<=100

5.3.2.2.6. ExternalReference

ExternalReference	
Name:	External reference
Definition:	Reference to an external information system containing any piece of information related to the spatial object.
Stereotypes:	«dataType»
Attribute: informationSystem	
Name:	Information system
Value type:	URI
Definition:	Uniform Resource Identifier of the external information system.
Multiplicity:	1
Attribute: informationSystemName	
Name:	Information system name
Value type:	PT_FreeText
Definition:	The name of the external information system.
Description:	EXAMPLES: Danish Register of Dwellings, Spanish Cadastre.
Multiplicity:	1
Attribute: reference	
Name:	Reference
Value type:	CharacterString
Definition:	Thematic identifier of the spatial object or of any piece of information related to the spatial object.
Description:	NOTE: This reference will act as a foreign key to implement the association between the spatial object in the INSPIRE data set and in the external information system. EXAMPLE: The cadastral reference of a given building in the national cadastral register.
Multiplicity:	1

5.3.2.3. Code lists

5.3.2.3.1. *ConditionOfConstructionValue*

ConditionOfConstructionValue	
Name:	Condition of construction value
Definition:	Values indicating the condition of a construction.
Extensibility:	none
Identifier:	http://inspire.ec.europa.eu/codelist/ConditionOfConstructionValue
Values:	The allowed values for this code list comprise only the values specified in <i>Annex C</i> .

5.3.2.3.2. *HeightStatusValue*

HeightStatusValue	
Name:	Height status value
Definition:	Values indicating the method used to capture a height.
Extensibility:	none
Identifier:	http://inspire.ec.europa.eu/codelist/HeightStatusValue
Values:	The allowed values for this code list comprise only the values specified in <i>Annex C</i> .

5.3.2.3.3. *ElevationReferenceValue*

ElevationReferenceValue	
Name:	Elevation reference value
Definition:	List of possible elements considered to capture a vertical geometry.
Description:	NOTE: The values of this code list are used to describe the reference of elevation both where elevation has been captured as attribute or as Z coordinate.
Extensibility:	none
Identifier:	http://inspire.ec.europa.eu/codelist/ElevationReferenceValue
Values:	The allowed values for this code list comprise only the values specified in <i>Annex C</i> .

5.3.2.3.4. *CurrentUseValue*

CurrentUseValue	
Name:	Current use value
Definition:	List of possible values indicating the current use.
Description:	SOURCE: This code list is partly based on and adapted from the Eurostat classification of types of constructions (for the classification of residential buildings). NOTE: the values of this code list apply to buildings or building components where building components may be a building part (in core profiles) or a building unit (in extended profiles)
Extensibility:	narrower
Identifier:	http://inspire.ec.europa.eu/codelist/CurrentUseValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and narrower values defined by data providers.

5.3.2.3.5. *BuildingNatureValue*

BuildingNatureValue	
Name:	Building nature value
Definition:	Values indicating the nature of a building.
Description:	NOTE 1 : This code list does not aim to be exhaustive as the attribute buildingNature addresses only noticeable buildings. NOTE 2: The values included in this code list address mainly (but not only) two international use cases: air flights where buildings may be obstacles and marine navigation where buildings may be landmarks.

BuildingNatureValue	
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/BuildingNatureValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers.

5.3.2.3.6. *HorizontalGeometryReferenceValue*

HorizontalGeometryReferenceValue	
Name:	Horizontal geometry reference value
Definition:	Values indicating the element considered to capture a horizontal geometry.
Extensibility:	none
Identifier:	http://inspire.ec.europa.eu/codelist/HorizontalGeometryReferenceValue
Values:	The allowed values for this code list comprise only the values specified in <i>Annex C</i> .

5.3.2.4. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

5.3.2.4.1. *Boolean*

Boolean	
Package:	Truth
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.3.2.4.2. *CharacterString*

CharacterString	
Package:	Text
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.3.2.4.3. *DateTime*

DateTime	
Package:	Date and Time
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.3.2.4.4. *DirectPosition*

DirectPosition	
Package:	Coordinate geometry
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]

5.3.2.4.5. *GM_Object*

GM_Object (abstract)	
Package:	Geometry root
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]

5.3.2.4.6. *GeographicalName*

GeographicalName	
Package:	Geographical Names
Reference:	INSPIRE Data specification on Geographical Names [DS-D2.8.I.3]
Definition:	Proper noun applied to a real world entity.

5.3.2.4.7. *Identifier*

Identifier	
-------------------	--

Identifier	
Package:	Base Types
Reference:	INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5]
Definition:	External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.
Description:	<p>NOTE1 External object identifiers are distinct from thematic object identifiers.</p> <p>NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object.</p> <p>NOTE 3 The unique identifier will not change during the life-time of a spatial object.</p>

5.3.2.4.8. *Integer*

Integer	
Package:	Numerics
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.3.2.4.9. *Length*

Length	
Package:	Units of Measure
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.3.2.4.10. *PT_FreeText*

PT_FreeText	
Package:	Cultural and linguistic adaptability
Reference:	Geographic information -- Metadata -- XML schema implementation [ISO/TS 19139:2007]

5.3.2.4.11. *URI*

URI	
Package:	basicTypes
Reference:	Geographic information -- Geography Markup Language (GML) [ISO 19136:2007]

5.4 Application schema Buildings2D

5.4.1 Description

5.4.1.1. Narrative description

The Buildings 2D application schema inherits of the semantics of Buildings base application schema and defines the geometric representation of buildings and building parts, using the data type BuildingGeometry2D, also defined in the Buildings Base application schema.

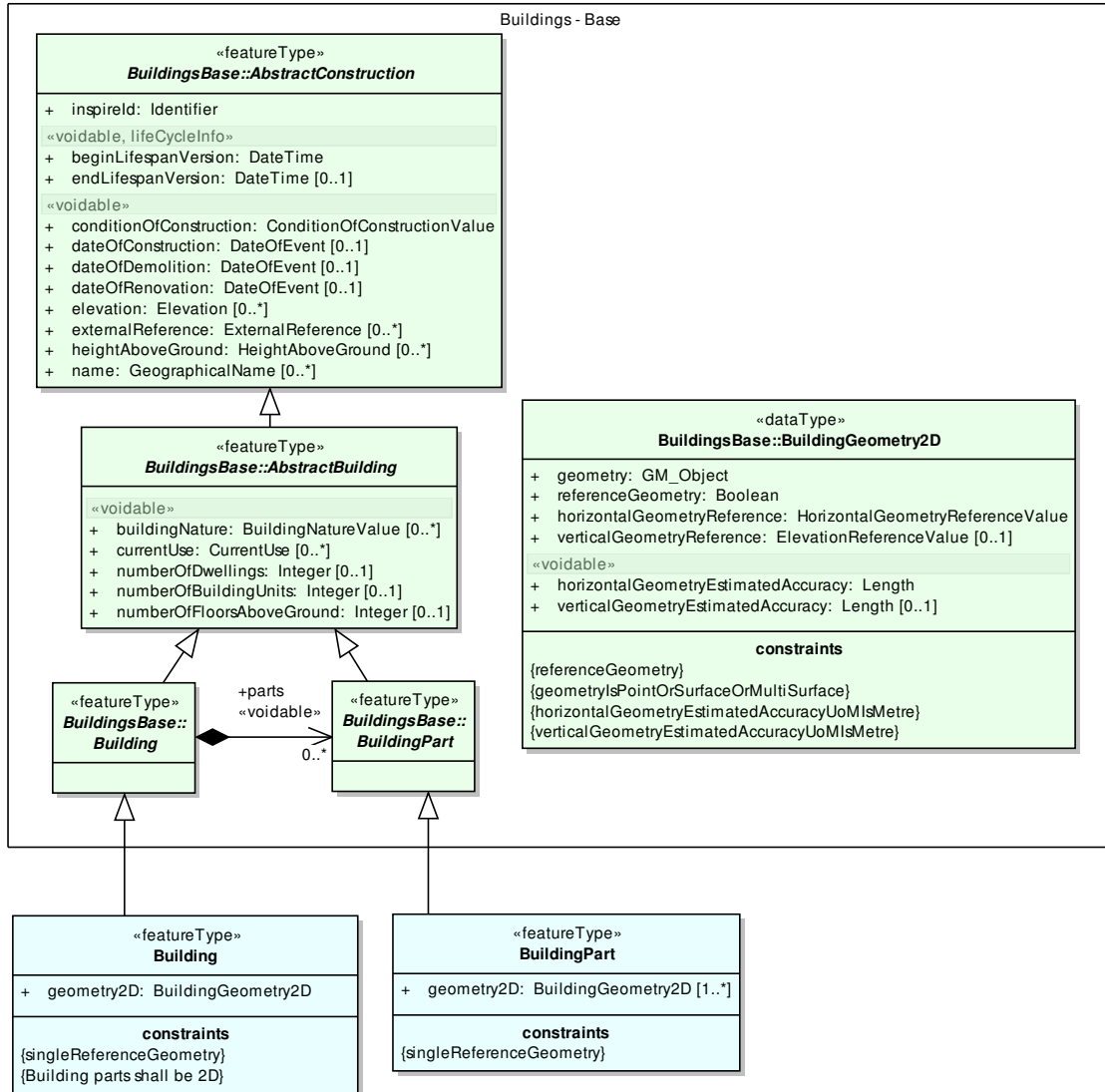


Figure 25: The Buildings 2D application schema

Multiple geometries are allowed for buildings; for instance, a data producer may provide representation of a building as a surface and as a point or as several surfaces, e.g. the building captured by its foot print and by its roof edges.

NOTE : the 2D application schema requires that both the geometry of the Building and of BuildingPart have to be provided (multiplicity [1..*]). In some cases, the value "combined" may be used to provide the horizontal geometry reference of Building, as shown in following illustration.

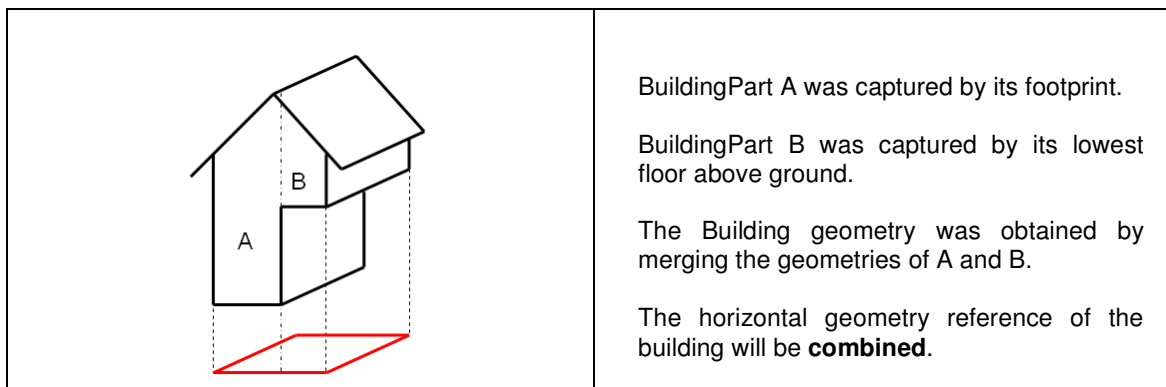


Figure 26: Example for value “combined”

5.4.1.2. UML Overview

See previous Figure 25: The Buildings 2D application schema.

5.4.1.3. Consistency between spatial data sets

The 2D building geometry may be used as reference geometry by governmental services in INSPIRE theme US; if this option is chosen by the data provider of US theme, this will ensure consistency between themes BU and US and will enable users to find a more detailed classification of the buildings hosting public services.

5.4.1.4. Geometry representation

The geometric representation of buildings and building parts has to be provided using the data type BuildingGeometry2D that is defined in <Buildings Base> application schema. It is reminded that this spatial properties allowed in this data types are restricted to Simple Feature v1.2.1 as defined by OGC document 06-103r4, i.e. to 0D, 1D, 2D and 2,5D data.

Recommendation 10 There should not be topological overlaps between buildings having same temporal validity.

NOTE 1: Topological overlaps are the overlaps which occur in the dataset without occurring in the real world, i.e. the overlaps due to bad quality of data.

NOTE 2: Overlaps may occur in the data set between buildings and/or building parts, due to the 2D (or 2,5D) representation of 3D real world objects.

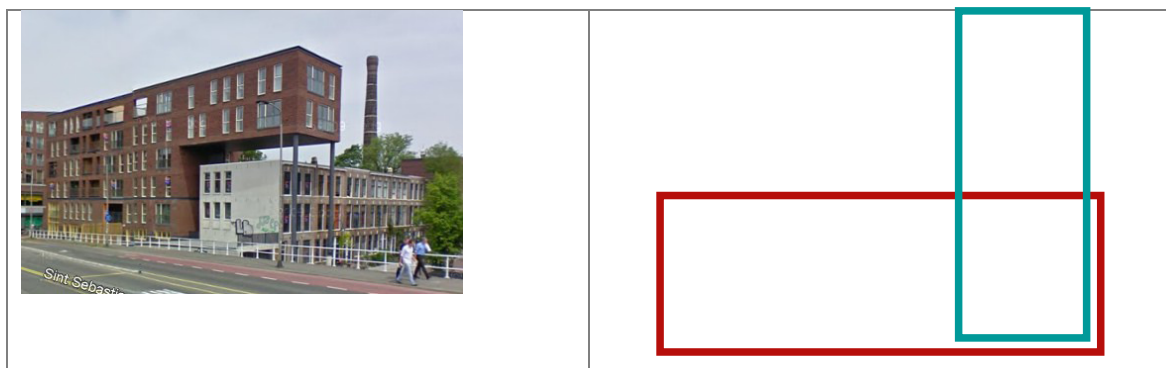


Figure 27: The 2D representations of these buildings are overlapping (this case of overlap is allowed)

Recommendation 1 The spatial objects Building should represent continuous or at least connected real world buildings, even if the representation may be done by a multi-surface.

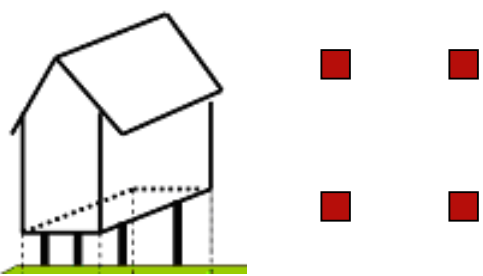
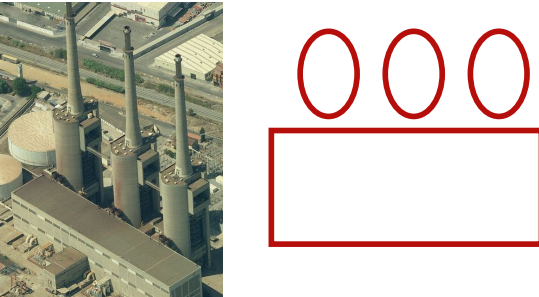
 A line drawing of a building on four piles. To the right of the building are four red squares arranged in a 2x2 grid, representing the footprint of the building.	 An aerial photograph of an industrial facility with a large rectangular block building and three tall cylindrical towers. To the right of the photo is a red outline consisting of three circles in a row above a large rectangle, representing a multi-surface representation.
<p>This continuous building on piles is represented by its footprint as a multisurface</p>	<p>The building is composed of a block building and of towers that are connected in real world but the building is represented as multi-surface, due to data capture rules.</p>

Figure 28: Examples where multi-surface may be used

5.4.2 Feature catalogue

Feature catalogue metadata

Application Schema	INSPIRE Application Schema Buildings2D
Version number	3.0

Types defined in the feature catalogue

Type	Package	Stereotypes
<i>Building</i>	Buildings2D	«featureType»
<i>BuildingPart</i>	Buildings2D	«featureType»

5.4.2.1. Spatial object types

5.4.2.1.1. Building

Building	
Name:	Building
Subtype of:	Building
Definition:	A Building is an enclosed construction above and/or underground, used or intended for the shelter of humans, animals or things or for the production of economic goods. A building refers to any structure permanently constructed or erected on its site.
Stereotypes:	«featureType»
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	BuildingGeometry2D
Definition:	2D or 2.5D geometric representation of the building.
Description:	NOTE: Multiple representations of the geometry are possible (e.g. by surface and by point).
Multiplicity:	1
Constraint: Building parts shall be 2D	
Natural language:	The parts of the building shall be represented using the BuildingPart type of the Buildings2D package.
OCL:	inv: self.parts->oclIsKindOf(Buildings2D::BuildingPart)
Constraint: singleReferenceGeometry	
Natural language:	Exactly one geometry2D attribute must be a reference geometry, i.e. the referenceGeometry attribute must be 'true'.
OCL:	inv: self.geometry2D->select(referenceGeometry=true)->size() = 1

5.4.2.1.2. BuildingPart

BuildingPart	
Name:	Building part
Subtype of:	BuildingPart
Definition:	A BuildingPart is a sub-division of a Building that might be considered itself as a building.
Description:	NOTE 1: A BuildingPart is homogeneous related to its physical, functional or temporal aspects. NOTE 2: Building and BuildingPart share the same set of properties. EXAMPLE: A building may be composed of two building parts having different heights above ground.
Stereotypes:	«featureType»

BuildingPart	
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	BuildingGeometry2D
Definition:	2D or 2.5D geometric representation of the building part.
Description:	NOTE: Multiple representations of the geometry are possible (e.g. by surface and by point).
Multiplicity:	1..*
Constraint: singleReferenceGeometry	
Natural language:	Exactly one geometry2D attribute must be a reference geometry, i.e. the referenceGeometry attribute must be 'true'.
OCL:	inv: self.geometry2D->select(referenceGeometry=true)->size() = 1

5.4.2.2. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

5.4.2.2.1. *BuildingGeometry2D*

BuildingGeometry2D	
Package:	BuildingsBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	This data types includes the geometry of the building and metadata information about which element of the building was captured and how.

5.5 Application schema Buildings3D

5.5.1 Description

5.5.1.1. Narrative description

The Buildings 3D application schema is a normative profile offered to data producers of 3D building data, in order to enable them to be INSPIRE conformant without having to "flatten" their data geometrically.

5.5.1.1.1. Feature types

The Buildings 3D application schema inherits of the semantics of <Buildings base> application schema and defines the geometric representation of buildings and building parts:

- The 3D representation has to be provided, using any of the LoD of City GML
- A 2D (or 2,5D) representation is allowed by the voidable attribute geometry2D

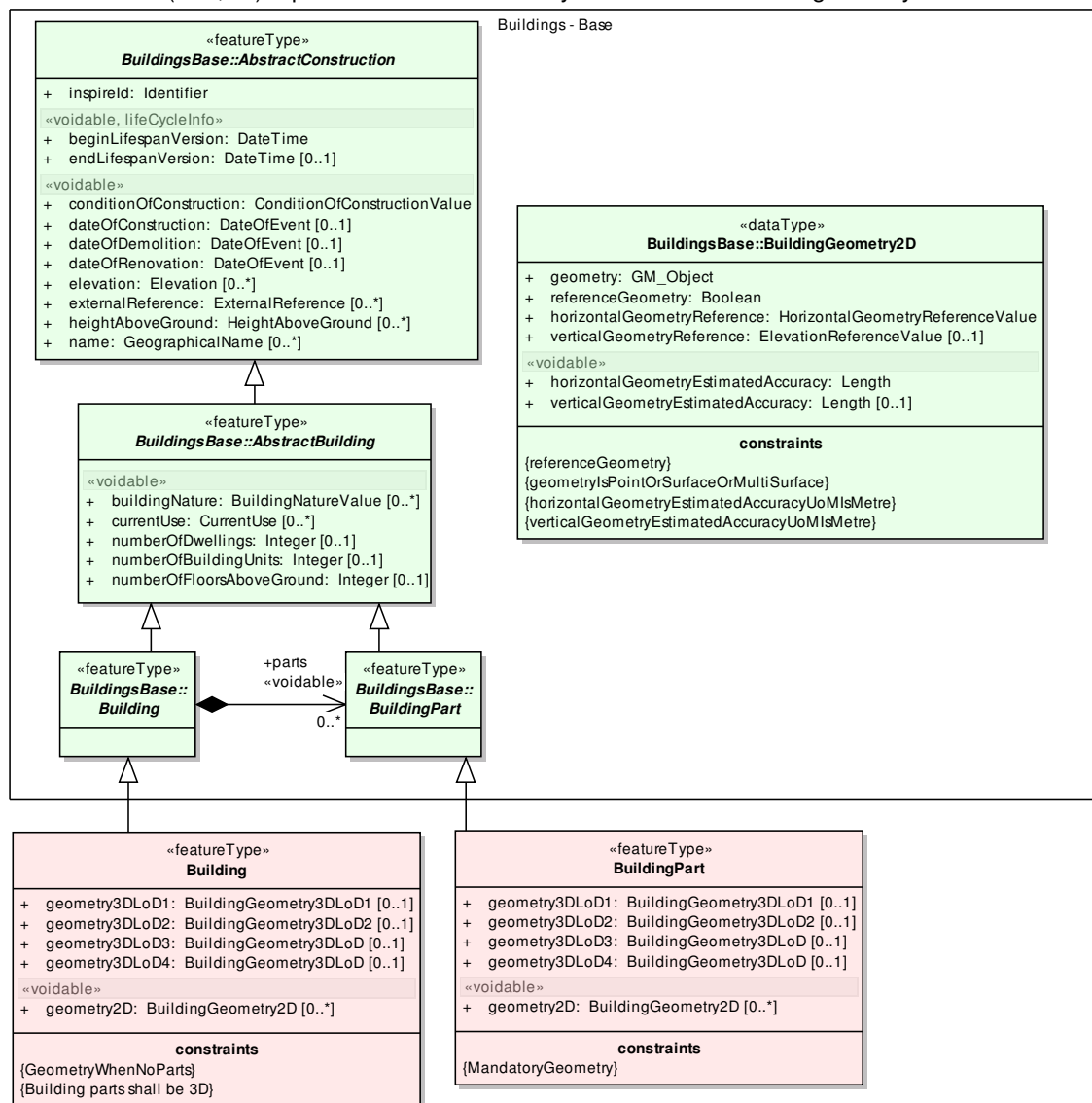


Figure 29: The Buildings 3D application schema

Buildings and building parts may be represented using any of the four levels of detail of City GML:

- a. in LoD1, a Building (or BuildingPart) is represented in a generalized way as right prism with vertical walls and horizontal 'roofs'. Such a model can be generated by vertically extruding a horizontal base polygon. It is often called "block model"
- b. in LoD2, a Building or BuildingPart is represented by a generalised way with vertical lateral surfaces and a prototypical roof or cover shape
- c. in LoD3 and LoD4, a Building or BuildingPart is represented by its real detailed shape for lateral faces (including protrusions, facade elements, and window recesses) as well as of the roof (including dormers, chimneys)

NOTE 1: The outer geometry of buildings in LoD3 and LoD4 is the same.

NOTE 2: the *Buildings3D* model allows to provide the four levels of City GML. However, it is very likely that data producers having LoD3 or LoD4 data will have more information about buildings than the content of this profile (e.g. description of the boundary surfaces, textures). In this case, the extended 3D profile will be more relevant.

The representation of *Buildings* in INSPIRE is based on the 4 levels of detail of City GML.

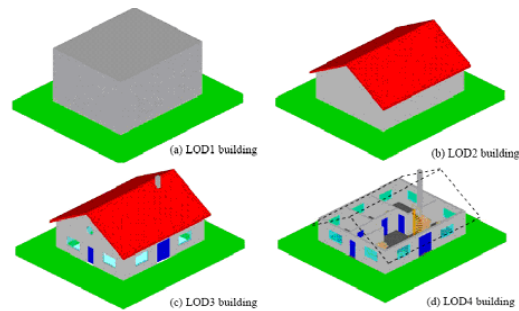


Figure 30: The 4 levels of detail of City GML

This means that it is possible, for instance:

- to choose one LoD and to use it for all buildings in the data set
- to use, in same data set, LoD1 for some buildings, LoD2 for some others and LoD3 or LoD4 for the last ones.
- to have, in same data set, several representations for the same building (e.g. one in LoD2 and one in LoD3)

Typically, in many existing data, ordinary buildings are represented using LoD1 or LoD2 whereas noticeable buildings or buildings in a project area will be represented with more details (LoD2, LoD3 or even LoD4).

The *Buildings3D* application schema only imposes that, at least, one of the mandatory geometries (i.e. LoD1, 2, 3 or 4 as solid or as multi-surface) is provided. This is indicated by a constraint.

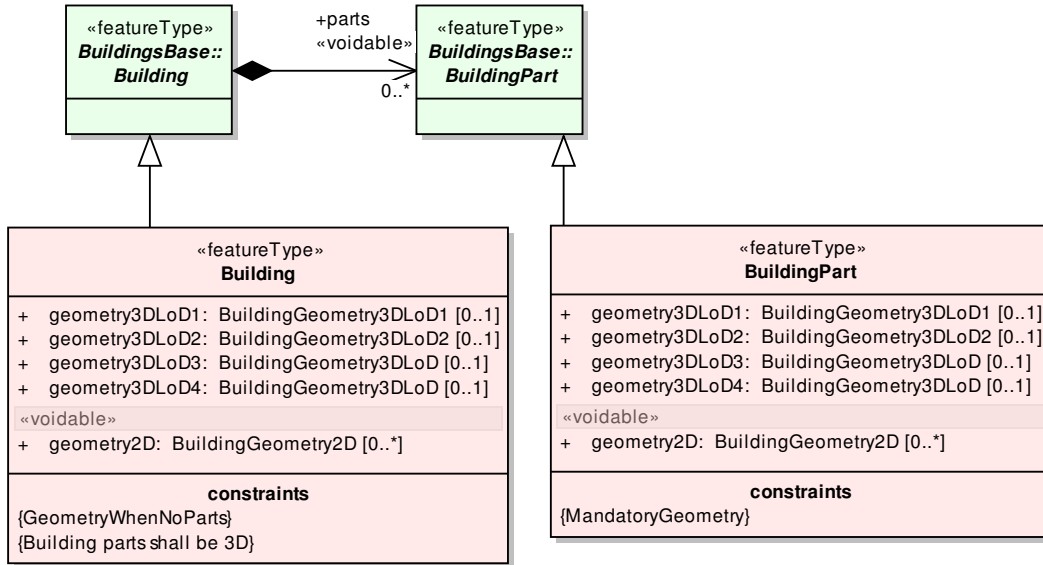


Figure 31: Geometry may be only on BuildingParts

In opposite to core 2D profile, duplication of geometry is not required in case of a building having building parts: the 3D geometry has to be provided on the building parts (constraint {MandatoryGeometry}) but is optional on the building. Of course, a simple building without any building part has a mandatory geometry (constraint {GeometryWhenNoParts}).

NOTE: the constraints “building parts shall be 3D” means that the building parts composing the buildings shall come to same application schema <Buildings 3D> and not from the <Buildings Base> application schema, i.e. the building parts shall have a 3D geometric representation.

5.5.1.1.2. Data types for 3D building geometry

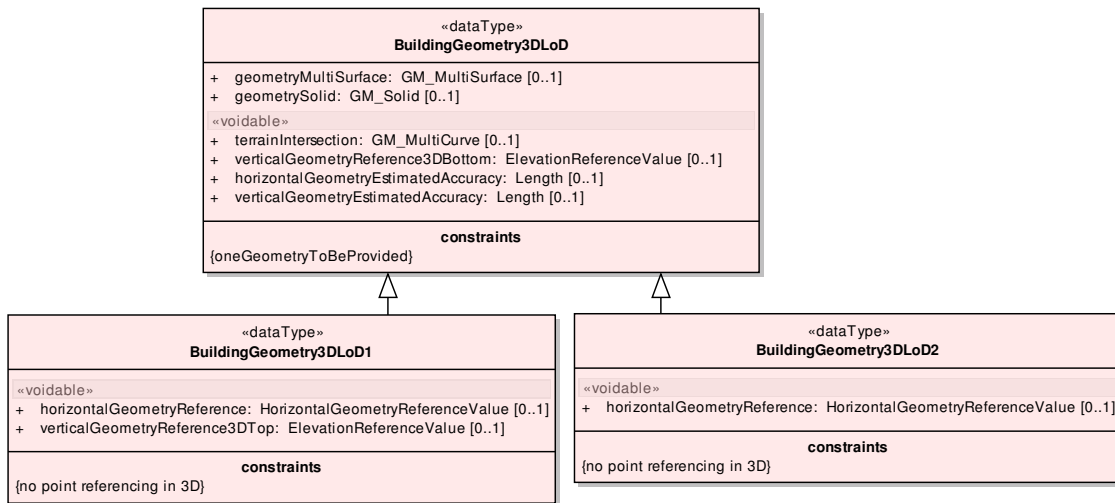


Figure 32: The data type BuildingGeometry3DLoD

For each level of detail of CityGML, the building or building part shall be represented either as a GM_Solid or as a GM_MultiSurface. If the representation as GM_Solid is chosen, the Building (or BuildingPart) is completely sealed by (non-overlapping) polygons in a topologically clean manner. This representation in general has the advantage that the volume can be computed. This is necessary, for

example, in disaster management applications to compute the volume of remaining breathing air or in environmental applications for the computation of energy related features. However, often such topologically clean models are not available in practice. This typically is the case if the data is captured by photogrammetric methods and particularly, the ground surface of the buildings (which is not observable by such methods) is missing. To accommodate for those models, the GM_MultiSurface representation is allowed.

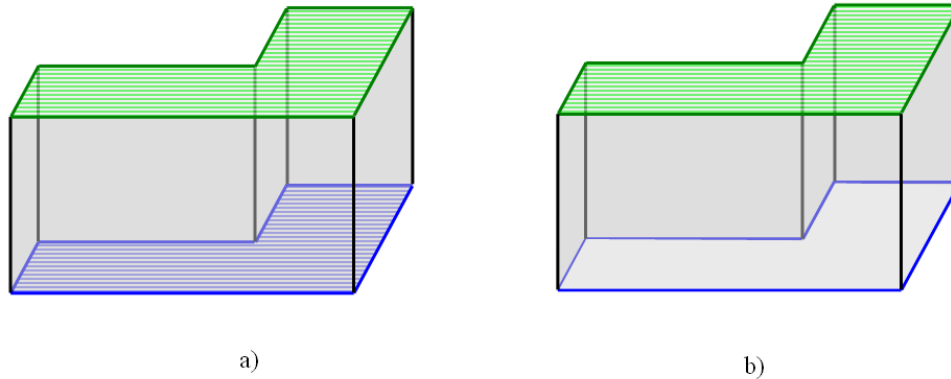


Figure 33: LoD 1 representation of a building as GM_Solid (a) and as GM_MultiSurface (b).

In both cases, the upper polygon (depicted hatched green) and the base geometry (blue) are horizontal. The side surfaces (grey) are rectangular and vertical. In the case of a GM_MultiSurface representation, the base polygon (depicted hatched blue) is missing.

In addition to the representation of a building by its outer shell in the four LoDs of City GML, the intersection of the building with the terrain may be provided, as a line.



Figure 34: The terrain intersection is shown in black

Moreover, the 3D geometry of the building or building part has to be documented.

For all LoDs, the level of building that was chosen to represent its bottom, has to be documented, through the attribute **verticalGeometryReference3DBottom** and using preferably the following values from the code list ElevationReferenceValue:

- generalGround
- lowestGroundPoint
- bottomOfConstruction
- lowestFloorAboveGround
- highestGroundPoint

Moreover, in case of LoD1 and LoD2, the representation of the building is only a generalised representation. So, as in core 2D profile, the **horizontal geometry reference** (that is the base for extrusion of the 3D geometry) has to be documented

The code list used to document the horizontal geometry reference is the same as in core 2D profile but the point references are not allowed (as a point horizontal geometry would not enable to represent the building as a volume). This is indicated by the constraint **{NoPointReferencingIn3D}**. In other words, only the values footprint, lowestFloorAboveGround, roofEdge, envelopeAboveGround and envelope may be used.

NOTE1: The horizontal geometry reference is not necessary for LoD3 and LoD4 where the 3D geometry shall represent the exact and detailed shape of the building. It is why this attribute is specific to LoD1 and LoD2.

NOTE 2: the value “combined” is not really suitable for 3D geometry, as the geometry of the building is optional when the building is the combination of several building parts.

Moreover, in case of LoD1 representation, the level of building that was chosen to represent its top, has to be documented, through the attribute **verticalGeometryReference3DTop** and using preferably the following values from the code list ElevationReferenceValue:

- generalRoofEdge
- lowestRoofEdge
- highestRoofEdge
- lowestEave
- generalEave
- highestEave
- generalRoof
- top OfConstruction

This information is not necessary for the other LoDs of City GML where the building or building part is represented with its roof. It is why this attribute is specific to LoD1.

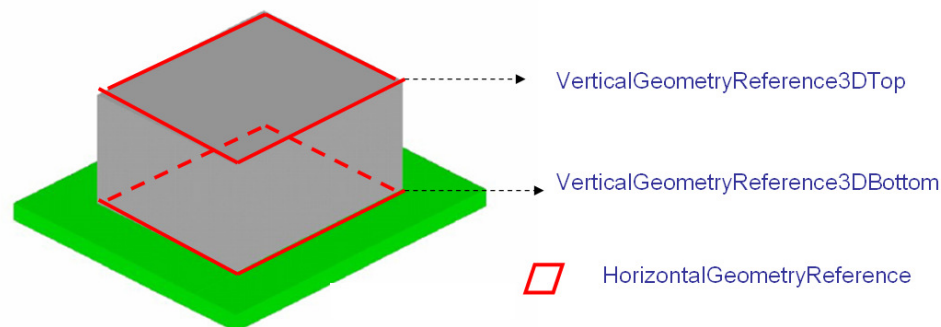


Figure 35: The 3D geometry of Building and BuildingPart has to be documented (example of LoD1)

5.5.1.2. UML Overview

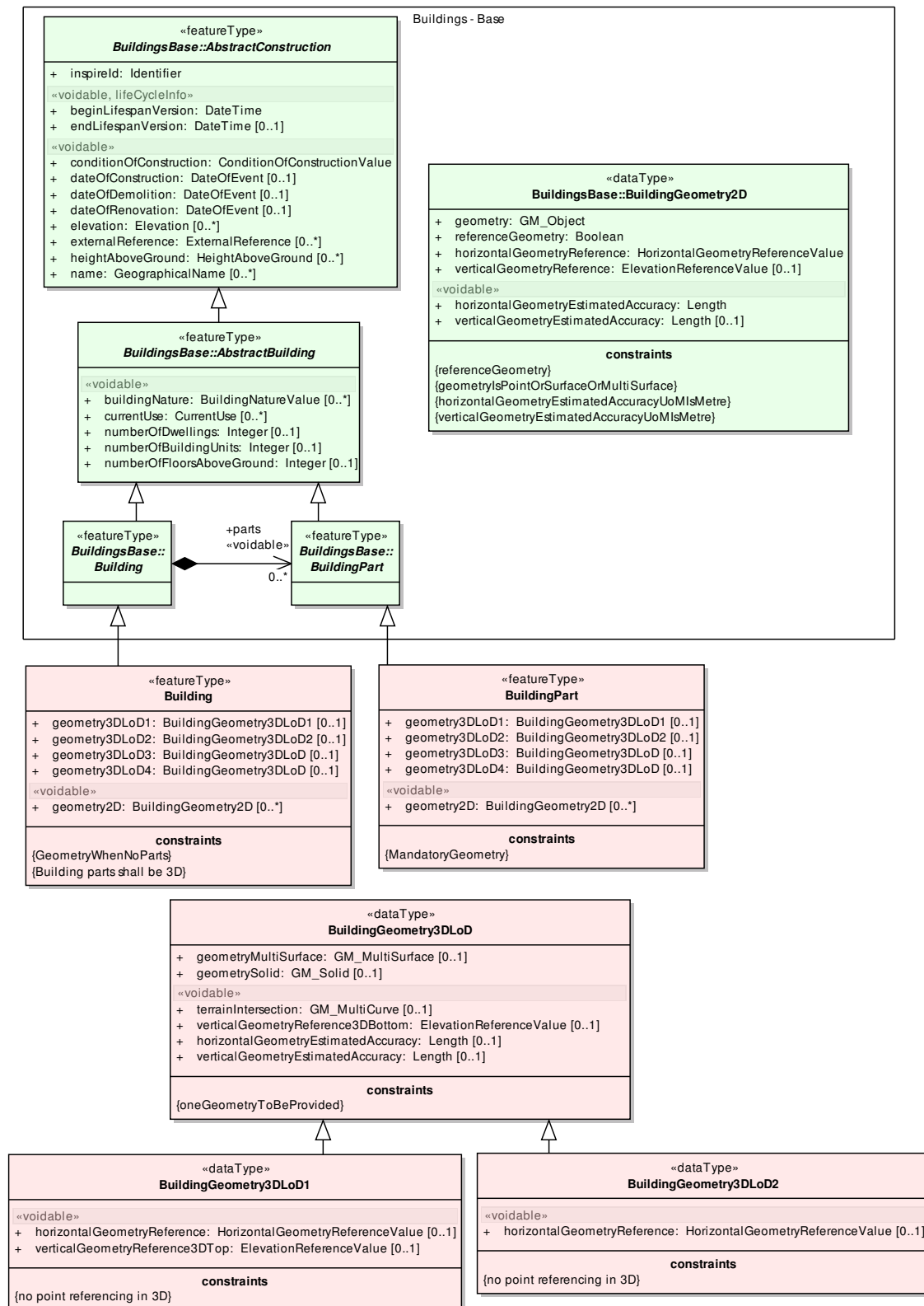


Figure 36: UML overview of Buildings Core 3D

5.5.1.3. Consistency between spatial data sets

It will be meaningful to use Buildings 3D data (when available) with INSPIRE themes taking also into account the vertical dimension, such as theme Elevation.

5.5.1.4. Modelling of object references

The external reference may be used as in core 2D profile. Moreover, in case the 2D (or 2,5D) representation of the building used to construct the 3D representation is not in the same data set as the 3D representation, the external reference may be used to link the spatial object in 3D data base to the 2D object representing the same building.

EXAMPLE: a local government has produced 3D data of buildings, using the 2D geometry provided by the national Cadastre. The external reference to the national cadastral system would enable users to know that these two related spatial objects represent the same real world building and would facilitate consistency between these various views on buildings.

5.5.1.5. Geometry representation

Art. 12(1) of Regulation 1089/2010 restricts the value domain of spatial properties to the Simple Feature spatial schema as defined in the *OpenGIS® Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture, version 1.2.1*, unless specified otherwise for a specific spatial data theme or type.

IR Requirement

Section 2.4

Theme-specific Requirement

By way of derogation from article 12(1), the value domain of spatial properties used in the *Buildings* 3D package shall not be restricted.

NOTE2: The topological relations of two spatial objects based on their specific geometry and topology properties can in principle be investigated by invoking the operations of the types defined in ISO 19107 (or the methods specified in EN ISO 19125-1).

Recommendation 11 There should not be penetration between buildings and/or building parts having same temporal validity.

NOTE: buildings and/or building parts may be touching (e.g. through common wall) but should not share common volume.

5.5.2 Feature catalogue

Feature catalogue metadata

Application Schema	INSPIRE Application Schema Buildings3D
Version number	3.0

Types defined in the feature catalogue

Type	Package	Stereotypes
<i>Building</i>	Buildings3D	«featureType»
<i>BuildingGeometry3DLoD</i>	Buildings3D	«data Type»
<i>BuildingGeometry3DLoD1</i>	Buildings3D	«data Type»
<i>BuildingGeometry3DLoD2</i>	Buildings3D	«data Type»
<i>BuildingPart</i>	Buildings3D	«featureType»

5.5.2.1. Spatial object types

5.5.2.1.1. Building

Building	
Name:	Building
Subtype of:	Building
Definition:	A Building is an enclosed construction above and/or underground, used or intended for the shelter of humans, animals or things or for the production of economic goods. A building refers to any structure permanently constructed or erected on its site.
Stereotypes:	«featureType»
Attribute: geometry2D	
Name:	geometry 2D
Value type:	BuildingGeometry2D
Definition:	2D or 2.5D geometric representation.
Description:	NOTE: Multiple representations of the geometry are possible (e.g. by surface and by point).
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: geometry3DLoD1	
Name:	geometry 3D LoD 1
Value type:	BuildingGeometry3DLoD1
Definition:	3D geometric representation at level of detail (LoD) 1, consisting of the generalized representation of the outer boundary by vertical lateral surfaces and horizontal base polygons.
Multiplicity:	0..1
Attribute: geometry3DLoD2	
Name:	geometry 3D LoD 2
Value type:	BuildingGeometry3DLoD2
Definition:	3D geometric representation at level of detail (LoD) 2, consisting of the generalized representation of the outer boundary by vertical lateral surfaces and a prototypical roof shape or cover (from a defined list of roof shapes)
Description:	NOTE: The prototypical roof shapes come from a defined list of roof shapes, in City GML; this list is equivalent to the code list RoofTypeValue, provided in the extended2D profile (without the hyperbolic paraboloidal roof).
Multiplicity:	0..1
Attribute: geometry3DLoD3	
Name:	geometry 3D LoD 3
Value type:	BuildingGeometry3DLoD3
Definition:	3D geometric representation at level of detail (LoD) 3, consisting of the detailed representation of the outer boundary (including protrusions, facade elements and window recesses) as well as of the roof shape (including dormers, chimneys).
Multiplicity:	0..1
Attribute: geometry3DLoD4	
Name:	geometry 3D LoD 4
Value type:	BuildingGeometry3DLoD4
Definition:	3D geometric representation at level of detail (LoD) 4, consisting of the detailed representation of the outer boundary (including protrusions, facade elements, and window recesses) as well as of the roof shape (including dormers, chimneys).

Building	
Description:	NOTE: The LoD4 representation is equivalent to the LoD3 representation in core 3D application schema. The LoD 4 representation is more meaningful in extended 3D application schema, with the optional description of building interior.
Multiplicity:	0..1
Constraint: Building parts shall be 3D	
Natural language:	The parts of the building shall be represented using the BuildingPart type of the Buildings3D package.
OCL:	inv: self.parts->oclIsKindOf(Buildings3D::BuildingPart)
Constraint: GeometryWhenNoParts	
Natural language:	If a Building does not have any BuildingParts, at least the geometry3DLoD1 or geometry3DLoD2 or geometry3DLoD3 or geometry3DLoD4 attributes shall be provided.
OCL:	

5.5.2.1.2. *BuildingPart*

BuildingPart	
Name:	Building part
Subtype of:	BuildingPart
Definition:	A BuildingPart is a sub-division of a Building that might be considered itself as a building.
Description:	NOTE 1: A building part is homogeneous related to its physical, functional and temporal aspects. EXAMPLE: A building may be composed of two building parts having different heights above ground.
Stereotypes:	«featureType»
Attribute: geometry2D	
Name:	geometry 2D
Value type:	BuildingGeometry2D
Definition:	2D or 2.5D geometric representation.
Description:	NOTE: Multiple representations of the geometry are possible (e.g. by surface and by point).
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: geometry3DLoD1	
Name:	geometry 3D LoD 1
Value type:	BuildingGeometry3DLoD1
Definition:	3D geometric representation at level of detail (LoD) 1, consisting of the generalized representation of the outer boundary by vertical lateral surfaces and horizontal base polygons.
Multiplicity:	0..1
Attribute: geometry3DLoD2	
Name:	geometry 3D LoD 2
Value type:	BuildingGeometry3DLoD2
Definition:	3D geometric representation at level of detail (LoD) 2, consisting of the generalized representation of the outer boundary by vertical lateral surfaces and a prototypical roof shape or cover (from a defined list of roof shapes). NOTE: The prototypical roof shapes come from a defined list of roof shapes, in City GML; this list is equivalent to the code list RoofTypeValue, provided in the

BuildingPart	
Multiplicity:	extended2D profile (without the hyperbolic paraboloidal roof). 0..1
Attribute: geometry3DLoD3	
Name:	geometry 3D LoD 3
Value type:	BuildingGeometry3DLoD
Definition:	3D geometric representation at level of detail (LoD) 3, consisting of the detailed representation of the outer boundary (including protrusions, facade elements and window recesses) as well as of the roof shape (including dormers, chimneys).
Multiplicity:	0..1
Attribute: geometry3DLoD4	
Name:	geometry 3D LoD 4
Value type:	BuildingGeometry3DLoD
Definition:	3D geometric representation at level of detail (LoD) 4, consisting of the detailed representation of the outer boundary (including protrusions, facade elements, and window recesses) as well as of the roof shape (including dormers, chimneys).
Description:	NOTE: The LoD4 representation is equivalent to the LoD3 representation in core 3D application schema. The LoD 4 representation is more meaningful in extended 3D application schema, with the optional description of building interior.
Multiplicity:	0..1
Constraint: MandatoryGeometry	
Natural language:	At least one of the geometry3DLoD1 or geometry3DLoD2 or geometry3DLoD3 or geometry3DLoD4 attributes shall be provided.
OCL:	

5.5.2.2. Data types

5.5.2.2.1. BuildingGeometry3DLoD

BuildingGeometry3DLoD	
Name:	Building geometry 3D LoD
Definition:	Data type grouping the 3D geometry of a building or building part and the metadata information attached to this geometry.
Stereotypes:	«dataType»
Attribute: geometryMultiSurface	
Name:	Geometry multi-surface
Value type:	GM_MultiSurface
Definition:	Representation of the outer boundary by a Multi Surface, which may - in contrast to a solid representation - not be topologically clean. In particular, the ground surface may be missing.
Multiplicity:	0..1
Attribute: geometrySolid	
Name:	Geometry solid
Value type:	GM_Solid
Definition:	Representation of the outer boundary by a solid.
Multiplicity:	0..1
Attribute: terrainIntersection	
Name:	Terrain intersection
Value type:	GM_MultiCurve
Definition:	Line or multi-line where the spatial object (Building, BuildingPart, ...) touches the terrain representation.

BuildingGeometry3DLoD	
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: verticalGeometryReference3DBottom	
Name:	Vertical geometry reference 3D bottom
Value type:	ElevationReferenceValue
Definition:	Height level to which the lower height of the model (Z-value of the lower horizontal polygon) refers to.
Description:	EXAMPLE: generalGround, bottomOfConstruction.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: horizontalGeometryEstimatedAccuracy	
Name:	Horizontal geometry estimated accuracy
Value type:	Length
Definition:	The estimated absolute positional accuracy of the (X,Y) coordinates of the geometry, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE: this mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: verticalGeometryEstimatedAccuracy	
Name:	Vertical geometry estimated accuracy
Value type:	Length
Definition:	The estimated absolute positional accuracy of the Z- coordinate of the geometry, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE: this mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Constraint: oneGeometryToBeProvided	
Natural language:	Either the geometryMultiSurface or the geometrySolid attribute shall be provided.
OCL:	inv: self.geometryMultiSurface->notEmpty() or self.geometrySolid->notEmpty()

5.5.2.2.2. BuildingGeometry3DLoD1

BuildingGeometry3DLoD1	
Name:	Building geometry 3D LoD 1
Subtype of:	BuildingGeometry3DLoD
Definition:	Data type grouping the specific metadata attached to the 3D geometry, when provided by a LoD 1 representation.
Stereotypes:	«dataType»
Attribute: horizontalGeometryReference	

BuildingGeometry3DLoD1	
Name:	Horizontal geometry reference
Value type:	HorizontalGeometryReferenceValue
Definition:	Element of the real world object that was captured by the (X,Y) coordinates of the LoD1 Multisurface or Solid geometry.
Description:	EXAMPLE: footprint, roof edge
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: verticalGeometryReference3DTop	
Name:	Vertical geometry reference 3D top
Value type:	ElevationReferenceValue
Definition:	Height level to which the upper height of the model (Z-value of the upper horizontal polygon) refers to.
Description:	EXAMPLE: generalRoof, lowestRoof Edge.
Multiplicity:	0..1
Stereotypes:	«voidable»
Constraint: no point referencing in 3D	
Natural language:	The horizontalGeometryReference attribute shall not take the value entrancePoint, pointInsideBuilding or pointInsideCadastralParcel.
OCL:	inv: self.horizontalGeometryReference->excludesAll(Set{HorizontalGeometryReferenceValue::entrancePoint, HorizontalGeometryReferenceValue::pointInsideBuilding, HorizontalGeometryReferenceValue::pointInsideCadastralParcel})

5.5.2.2.3. BuildingGeometry3DLoD2

BuildingGeometry3DLoD2	
Name:	Building geometry 3D LoD 2
Subtype of:	BuildingGeometry3DLoD
Definition:	Data type grouping the specific metadata attached to the 3D geometry, when provided by a LoD2 representation.
Stereotypes:	«dataType»
Attribute: horizontalGeometryReference	
Name:	Horizontal geometry reference
Value type:	HorizontalGeometryReferenceValue
Definition:	Element that was captured by the (X,Y) coordinates of the LoD2 MultiSurface or Solid geometry.
Description:	EXAMPLE: footprint, roof edge
Multiplicity:	0..1
Stereotypes:	«voidable»
Constraint: no point referencing in 3D	
Natural language:	The horizontalGeometryReference attribute shall not take the value entrancePoint, pointInsideBuilding or pointInsideCadastralParcel.
OCL:	inv: self.horizontalGeometryReference->excludesAll(Set{HorizontalGeometryReferenceValue::entrancePoint, HorizontalGeometryReferenceValue::pointInsideBuilding, HorizontalGeometryReferenceValue::pointInsideCadastralParcel})

5.5.2.3. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

5.5.2.3.1. *BuildingGeometry2D*

BuildingGeometry2D

Package:	BuildingsBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	This data types includes the geometry of the building and metadata information about which element of the building was captured and how.

5.5.2.3.2. *ElevationReferenceValue*

ElevationReferenceValue

Package:	BuildingsBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	List of possible elements considered to capture a vertical geometry.
Description:	NOTE: The values of this code list are used to describe the reference of elevation both where elevation has been captured as attribute or as Z coordinate.

5.5.2.3.3. *GM_MultiCurve*

GM_MultiCurve

Package:	Geometric aggregates
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]

5.5.2.3.4. *GM_MultiSurface*

GM_MultiSurface

Package:	Geometric aggregates
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]

5.5.2.3.5. *GM_Solid*

GM_Solid

Package:	Geometric primitive
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]

5.5.2.3.6. *HorizontalGeometryReferenceValue*

HorizontalGeometryReferenceValue

Package:	BuildingsBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Values indicating the element considered to capture a horizontal geometry.

5.5.2.3.7. *Length*

Length

Package:	Units of Measure
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.6 Application schema BuildingsExtendedBase

5.6.1 Description

5.6.1.1. Narrative description

Buildings Base Extended is an abstract application schema describing the additional semantics that is common to instanciable application schemas Buildings Extended2D and Buildings Extended3D.

5.6.1.1.1. Additional feature types

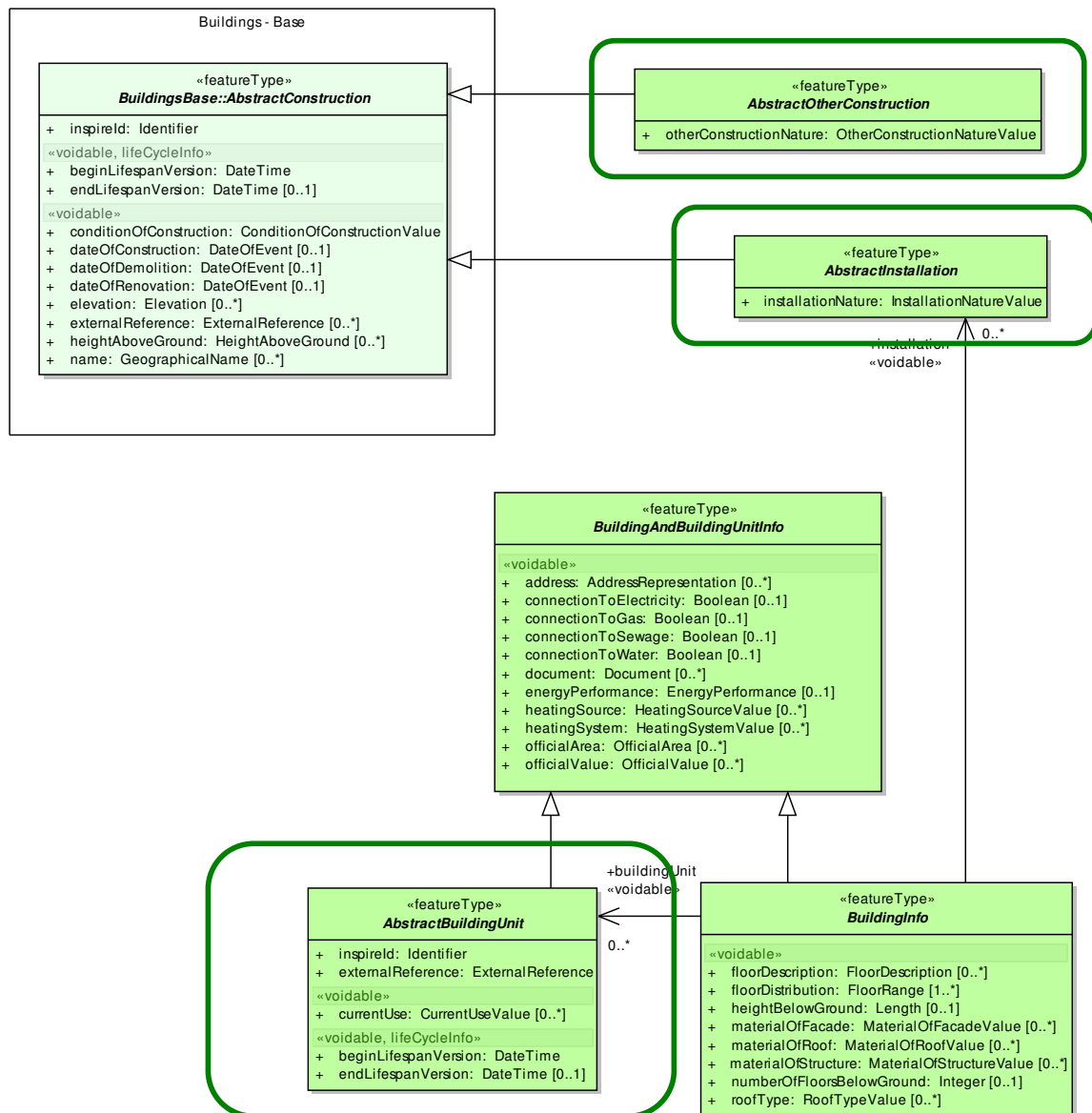


Figure 37: Main feature types of Buildings Base Extended

The Buildings BaseExtended contains mainly 3 new feature types:

- OtherConstructions are self-standing constructions that are generally not considered as buildings. This extended profile includes the most significant constructions that are necessary to describe landscape and to fulfil use cases such as safety or spatial planning.
- Installations are constructions, generally of small size that are attached to a Building (or a BuildingPart).
- BuildingUnits are subdivisions of Building with their own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which are atomic, functionally independent, and may be separately sold, rented out, inherited, etc.

A building unit is homogeneous for management aspects. Its key mandatory attribute is the **external reference** to some official register where the BuildingUnit is identified and described. It is generally the cadastral register but may be another information system, e.g. a register of public properties.

5.6.1.1.2. Other Constructions

Other constructions inherit of the attributes of AbstractConstruction (from <BuildingsBase>) and are mainly described by their nature, that may take following values:













				
acousticFence	antenna	chimney	bridge	bridge
				
cityWall	crane	monument	monument	monument
				
monument	monument	openAirPool	protectiveStructure	protectiveStructure
				
pylon	retainingWall	solarPanel	substation	tunnel

Figure 38: Illustrations of other constructions

5.6.1.1.3. Installations

Installations inherit of the attributes of AbstractConstruction (from <BuildingsBase>) and are mainly described by their nature, that may take the following values.





















				
airConditioningUnit	airDuct	antenna	antenna	arcade
				
balcony	chimney	chimney	cradle	dormer
				
externalLift	railing	ramp	solarPanel	stairway
				
stairway	stairway	tower	windTurbine	windTurbine

Figure 39: Illustrations for of installation nature

NOTE 1: The list of installation nature does not aim to be exhaustive but focus on the installations related to safety and on environmental issues (mainly energy).

NOTE 2: The code list for installation nature is the same for extended 2D and extended 3D data. However, likely, some values will be used only for 3D data (e.g. dormer, arcade, balcony).

5.6.1.1.4. New properties

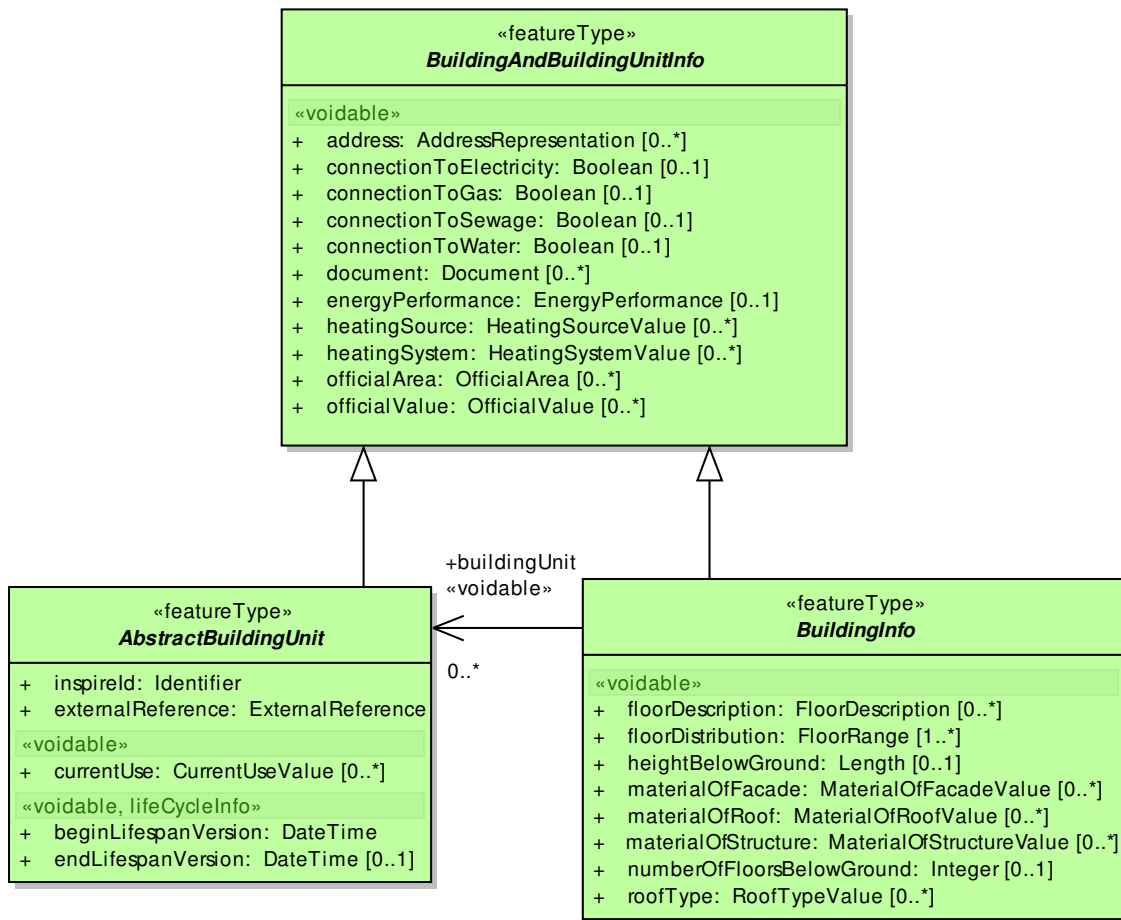


Figure 40: The new properties of buildings / building parts

Buildings BaseExtended defines additional properties that will apply to buildings and building parts in application schema Extended2D and Extended3D.

These additional properties are gathered in two abstract feature types BuildingAndBuildingUnitInfo and BuildingInfo.

NOTE: these two new classes are just container for the additional attributes and associations rather than real feature types.

- The additional properties that are common to buildings, building parts and building units are grouped in feature type BuildingAndBuildingUnitInfo. These common attributes are mainly related to official information. In addition to the common attributes shown in the figure below, BuildingAndBuildingUnitInfo has associations to feature types Address and Cadastral Parcels that are in annex I themes.

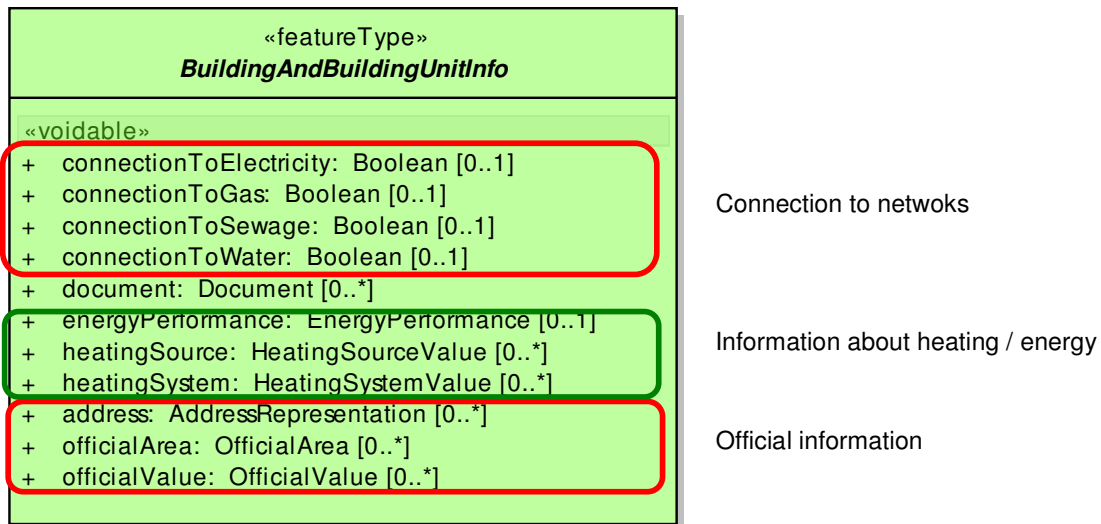


Figure 41: Common additional attributes of buildings, building parts and building units

- The additional properties that are specific to buildings and building parts are grouped in feature type BuildingInfo. These attributes address the physical description with more details than in the <Buildings Base> application schema.

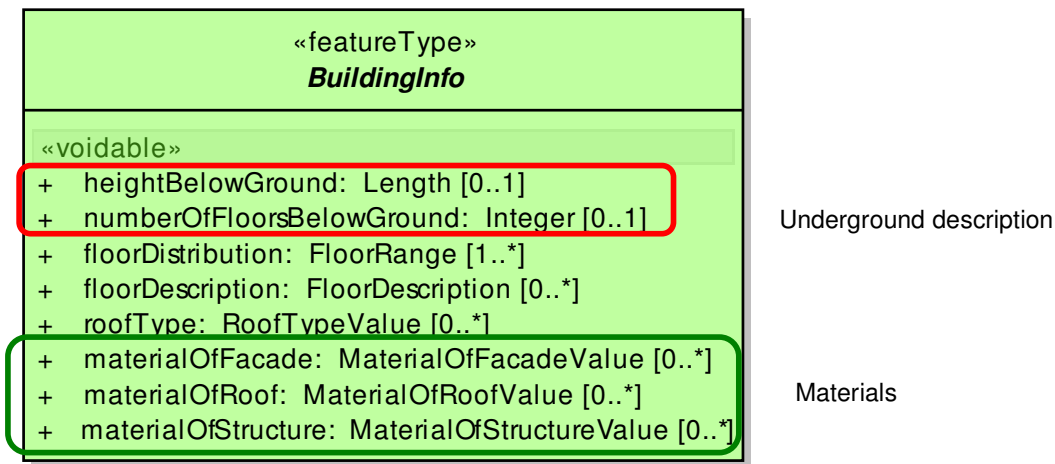
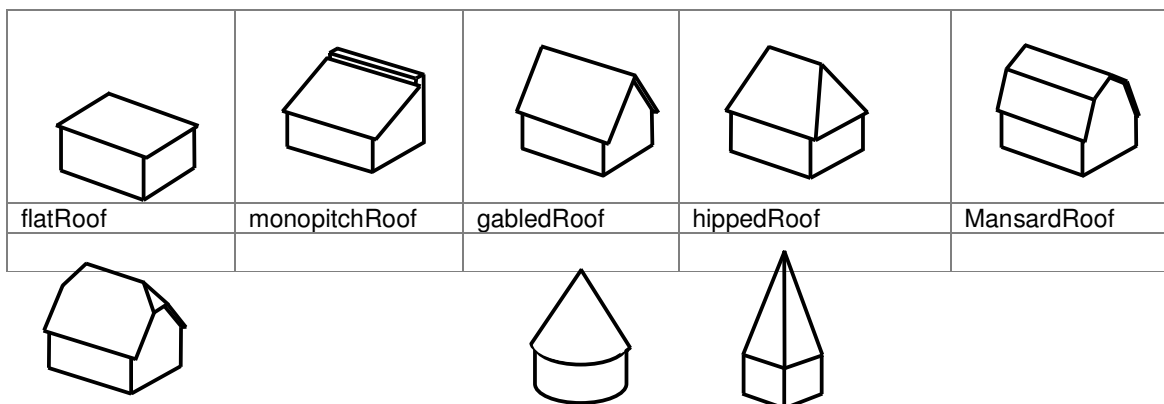


Figure 42: Common additional attributes to buildings and building parts

5.6.1.1.5. Attribute roofType

This attribute may take the following values:





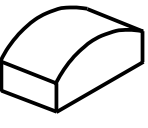
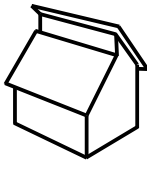

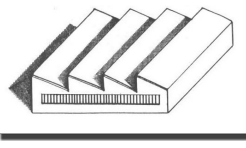
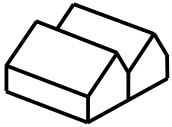
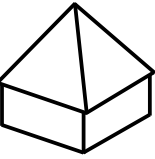


				
halfHippedRoof		coneRoof	pyramidalBroachRoof	copulaRoof
				
archRoof	dualPentRoof		sawToothRoof	
				
pavilionRoof	hyperbolicParabaloidalRoof			

Figure 43: Roof types (most illustrations from 3D GIS)

5.6.1.1.6. *Attribute MaterialOfStructure*

This attribute may take the following values:

			
adobeBlockWalls	concreteBlockMasonry	earth	firedBrickMasonry
			
informalConstructions	massiveStoneMasonry	mobileHomes	mudWalls
			
precastConcrete Tilt-upWalls	reinforcedConcrete	reinforcedMasonry	rubleStoneMasonry

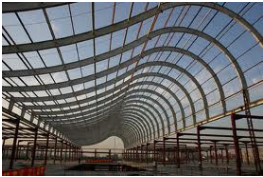



			
steel	stoneMasonryBlock	wood	wood

Figure 44: Illustrations for material of structure

5.6.1.1.7. *MaterialOfFacade*

This attribute may take the following values:













			
adobe	asbestos	ceramicTiles	composite
			
concrete	glass	limestone	masonry
			
metal	naturalStone	vegetated	wood

Figure 45: Illustrations for material of facade

5.6.1.1.8. *MaterialOfRoof*

This attribute may take the following values:


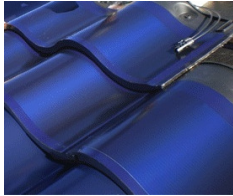

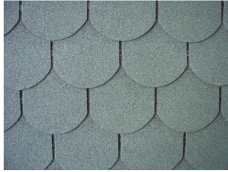

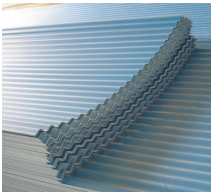










			
asbestos	ceramicTiles	clayTile	composition
			
concreteTile	corrugatedSheet	glass	hotMoppedAsphalt
			
metal	reinforcedConcrete	slate	slate
			
thatch	thatch	vegetatedRoof	woodShingles OrShakes

Figure 46: Illustrations for material of roof

5.6.1.1.9. *Attribute Document*

The INSPIRE model allows the possibility to link documents to a building or a building part or a floor or a building unit; various documents may be concerned, such as images, sketches, building permits, emergency plans The attribute Document is defined as a data type with the link to the place the document may be found and with a simple set of metadata elements.

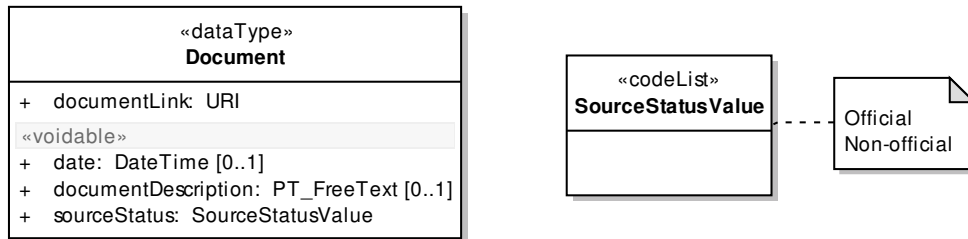


Figure 47: The data type Document

Recommendation 1 Documents should be provided in well-known and easy to handle formats.

EXAMPLE: documents may be provided in .PDF, .TIF (or geotif), .JPEG, .BMP, .PNG.

NOTE 1: **Formats whose content is unknown to user, such as .EXE or .ZIP should be avoided.**

NOTE 2: **The documents related to the regulations that apply on all buildings in an area of interest (e.g. land use zone, regulated area, protected site) may and should rather be provided in the respective other INSPIRE themes.**

5.6.1.1.10. Attribute officialValue

The attribute official value may be submitted to access restrictions due for instance to privacy issues. Consequently, the INSPIRE model allows this information to be provided, either directly by its value and currency or indirectly by the external reference to another information system. In first case, the access to the information of official area is widely open, following INSPIRE rules. In the second case, the access to the other external information system may be restricted to authorized users. In addition to the official value itself, some metadata elements should be provided in order to indicate what the official value represents.

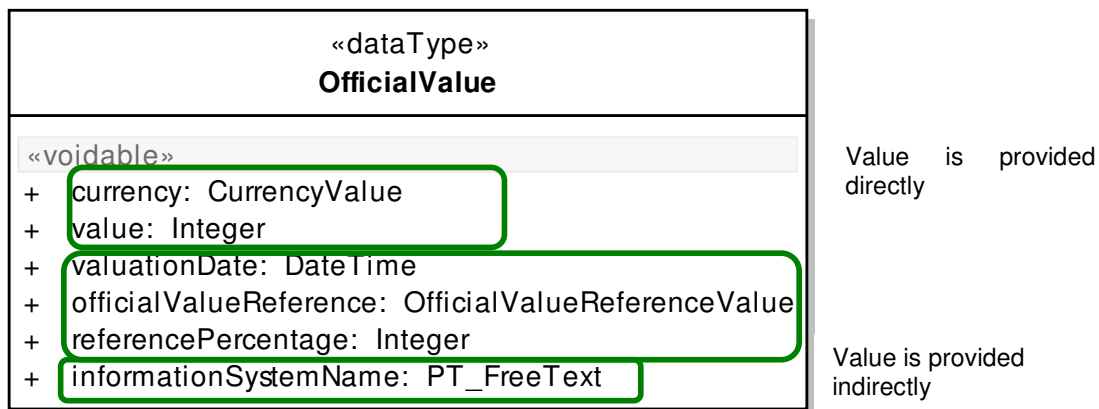


Figure 48: The data type OfficialValue

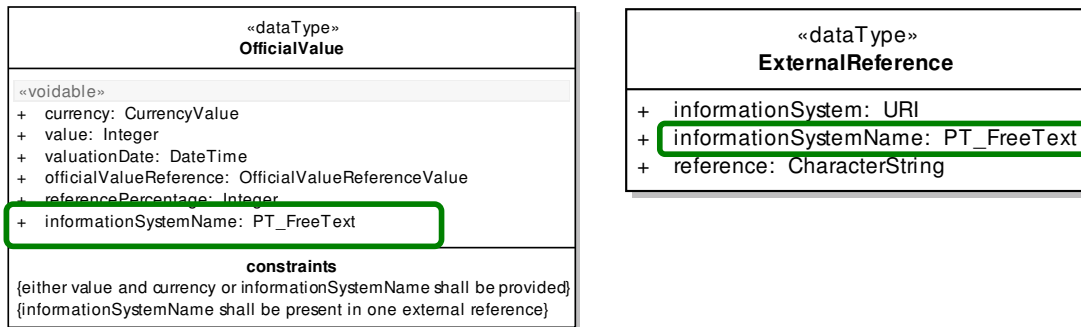


Figure 49: The mechanism to get official value through external reference to another information

NOTE: The mechanism to provide value of an attribute either directly or through the reference to another information system may be used by data providers for other attributes, in case they are submitted to access restrictions.

5.6.1.2. UML Overview

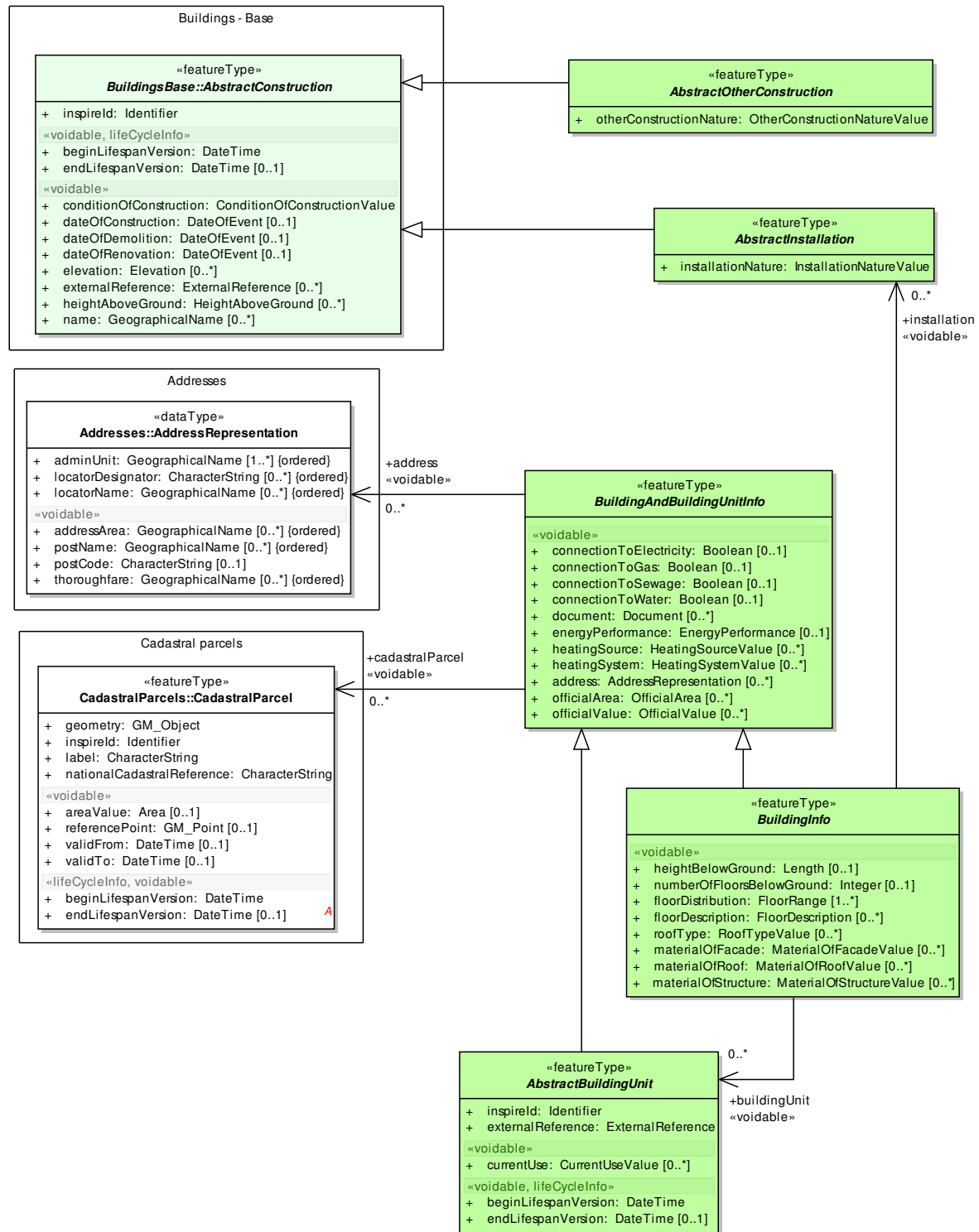


Figure 50: Overview of BuildingExtendedBase

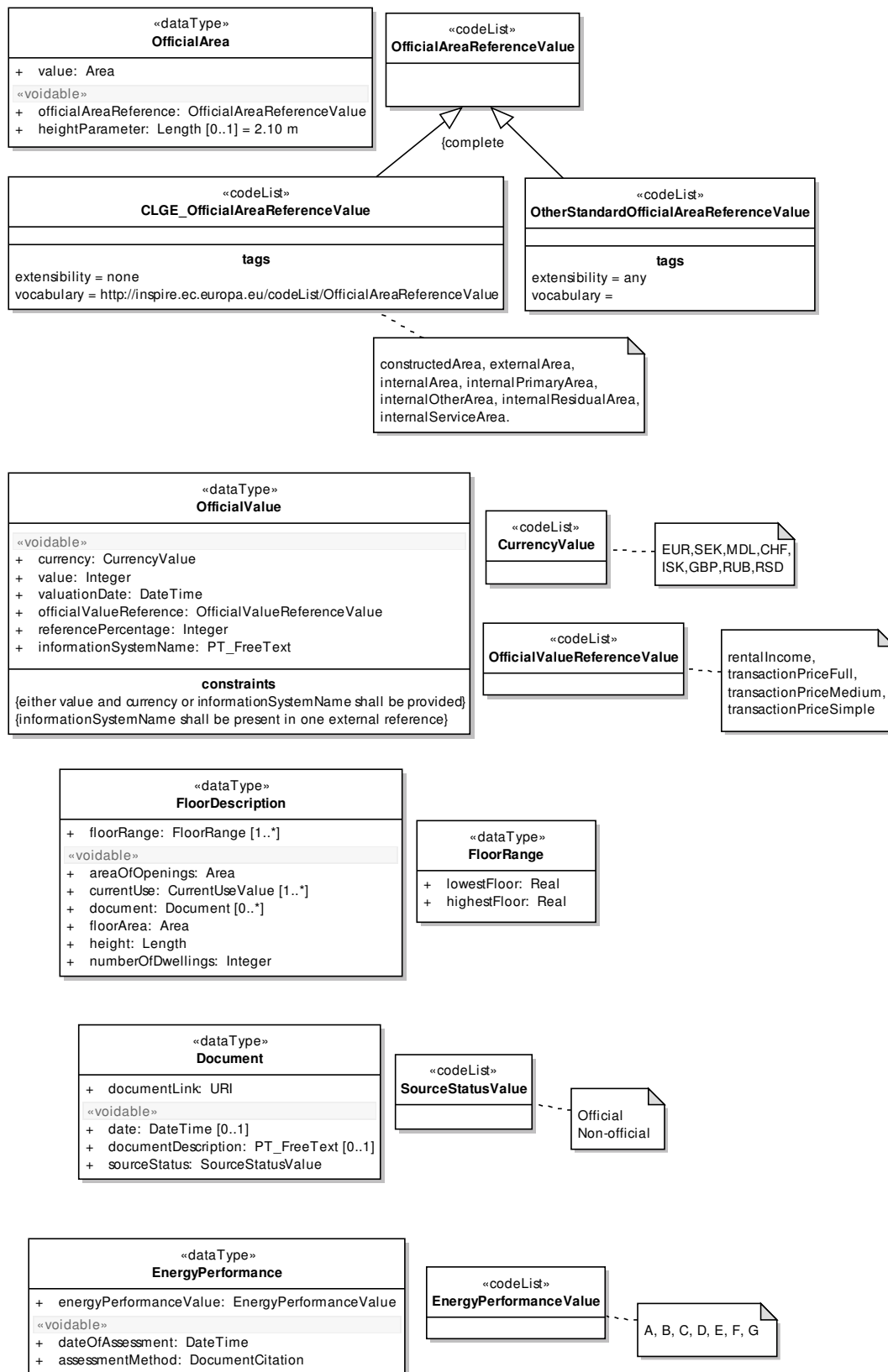


Figure 51: Overview of BuildingExtendedBase - Data Types

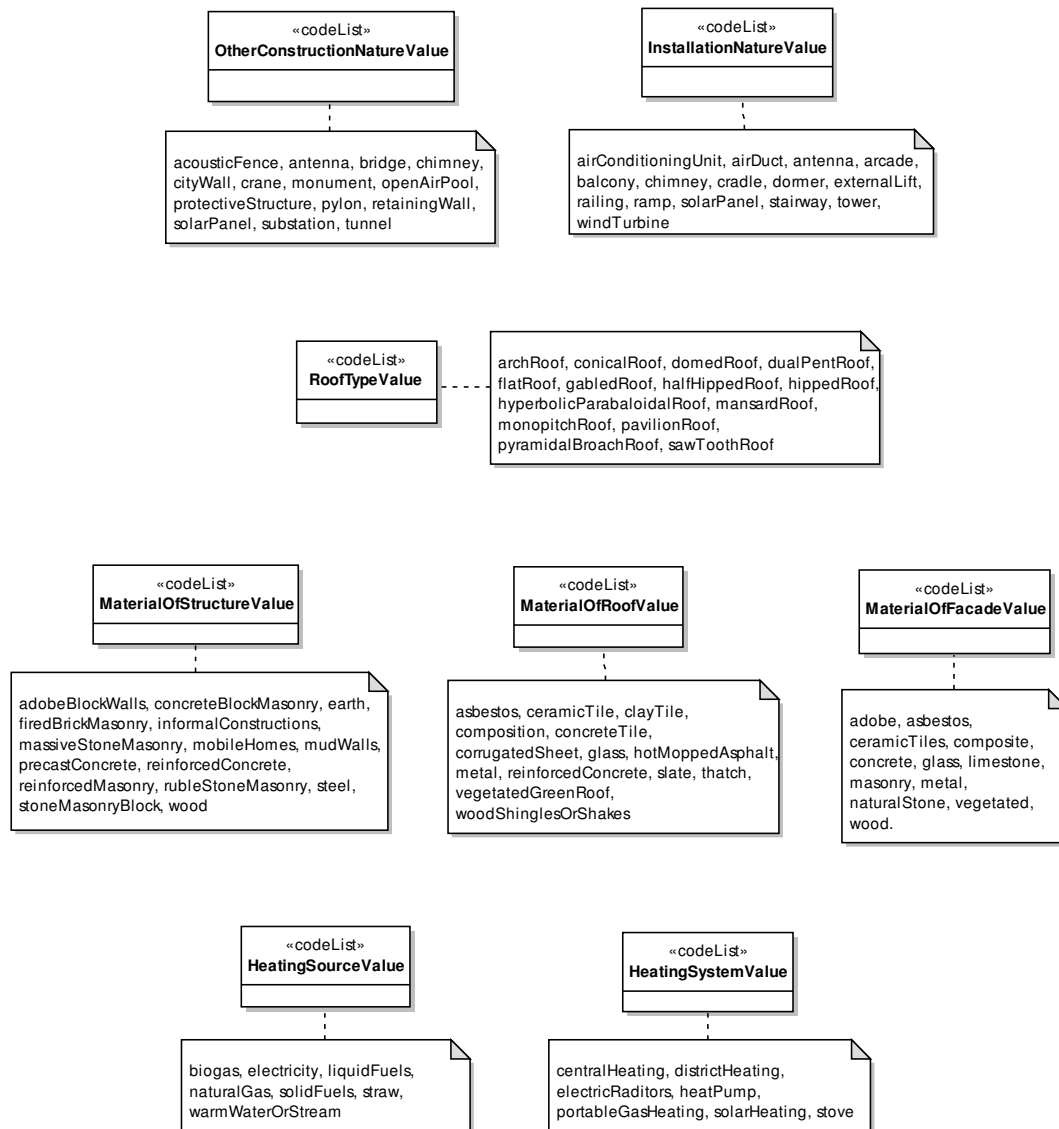


Figure 52: Overview of BuildingsExtendedBase - Code lists

5.6.2 Feature catalogue

Feature catalogue metadata

Application Schema	INSPIRE Application Schema BuildingsExtendedBase
Version number	3.0

Types defined in the feature catalogue

Type	Package	Stereotypes
<i>AbstractBuildingUnit</i>	BuildingsExtendedBase	«featureType»
<i>AbstractInstallation</i>	BuildingsExtendedBase	«featureType»
<i>AbstractOtherConstruction</i>	BuildingsExtendedBase	«featureType»
<i>BuildingAndBuildingUnitInfo</i>	BuildingsExtendedBase	«featureType»
<i>BuildingInfo</i>	BuildingsExtendedBase	«featureType»
<i>CLGE_OfficialAreaReferenceValue</i>	BuildingsExtendedBase	«codeList»

Type	Package	Stereotypes
<i>CurrencyValue</i>	BuildingsExtendedBase	«codeList»
<i>Document</i>	BuildingsExtendedBase	«dataType»
<i>EnergyPerformance</i>	BuildingsExtendedBase	«dataType»
<i>EnergyPerformanceValue</i>	BuildingsExtendedBase	«codeList»
<i>FloorDescription</i>	BuildingsExtendedBase	«dataType»
<i>FloorRange</i>	BuildingsExtendedBase	«dataType»
<i>HeatingSourceValue</i>	BuildingsExtendedBase	«codeList»
<i>HeatingSystemValue</i>	BuildingsExtendedBase	«codeList»
<i>InstallationNatureValue</i>	BuildingsExtendedBase	«codeList»
<i>MaterialOfFacadeValue</i>	BuildingsExtendedBase	«codeList»
<i>MaterialOfRoofValue</i>	BuildingsExtendedBase	«codeList»
<i>MaterialOfStructureValue</i>	BuildingsExtendedBase	«codeList»
<i>OfficialArea</i>	BuildingsExtendedBase	«dataType»
<i>OfficialAreaReferenceValue</i>	BuildingsExtendedBase	«codeList»
<i>OfficialValue</i>	BuildingsExtendedBase	«dataType»
<i>OfficialValueReferenceValue</i>	BuildingsExtendedBase	«codeList»
<i>OtherConstructionNatureValue</i>	BuildingsExtendedBase	«codeList»
<i>OtherStandardOfficialAreaReferenceValue</i>	BuildingsExtendedBase	«codeList»
<i>RoofTypeValue</i>	BuildingsExtendedBase	«codeList»
<i>SourceStatusValue</i>	BuildingsExtendedBase	«codeList»

5.6.2.1. Spatial object types

5.6.2.1.1. AbstractBuildingUnit

AbstractBuildingUnit (abstract)	
Name:	Abstract building unit
Subtype of:	BuildingAndBuildingUnitInfo
Definition:	Abstract spatial object type grouping the semantic properties of building units. A BuildingUnit is a subdivision of Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc.
Description:	Building units are spatial objects aimed at subdividing buildings and/or building parts into smaller parts that are treated as separate entities in daily life. A building unit is homogeneous, regarding management aspects. EXAMPLES: It may be e.g. an apartment in a condominium, a terraced house, or a shop inside a shopping arcade. NOTE 1: According to national regulations, a building unit may be a flat, a cellar, a garage or set of a flat, a cellar and a garage. NOTE 2: According to national regulation, a building that is one entity for daily life (typically, a single family house) may be considered as a Building composed of one BuildingUnit or as a Building composed of zero BuildingUnit.
Stereotypes:	«featureType»
Attribute: inspireId	
Name:	inspire id
Value type:	Identifier
Definition:	External object identifier of the spatial object.
Description:	An external object identifier is a unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object. The identifier is an identifier of the spatial object, not an identifier of the real-world phenomenon.
Multiplicity:	1

AbstractBuildingUnit (abstract)**Attribute: currentUse**

Name: Current use
Value type: CurrentUseValue
Definition: Activity hosted by the building unit.
Multiplicity: 0..*
Stereotypes: «voidable»

Attribute: externalReference

Name: External reference
Value type: ExternalReference
Definition: Reference to an external information system containing any piece of information related to the spatial object.
Description: Typically, the external reference will be established to the information system where BuildingUnits are defined.
EXAMPLES: the information system will be the cadastral register or an official dwelling register. It may be also a register of public properties.
Multiplicity: 1

Attribute: beginLifespanVersion

Name: Begin lifespan version
Value type: DateTime
Definition: Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
Multiplicity: 1
Stereotypes: «voidable,lifeCycleInfo»

Attribute: endLifespanVersion

Name: End lifespan version
Value type: DateTime
Definition: Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
Multiplicity: 0..1
Stereotypes: «voidable,lifeCycleInfo»

*5.6.2.1.2. AbstractInstallation***AbstractInstallation (abstract)**

Name: Abstract installation
Subtype of: AbstractConstruction
Definition: Abstract spatial object type grouping the semantic properties of installations. An external construction (of small size) or an external device serving the building or building part.
Description: EXAMPLES: stairway, solar panel, external lift
Stereotypes: «featureType»

Attribute: installationNature

Name: Installation nature
Value type: InstallationNatureValue
Definition: A description of the installation that represents its intended nature or current function.
Multiplicity: 1

*5.6.2.1.3. AbstractOtherConstruction***AbstractOtherConstruction (abstract)**

AbstractOtherConstruction (abstract)	
Name:	Abstract other construction
Subtype of:	AbstractConstruction
Definition:	Abstract spatial object type grouping the semantic properties of other constructions. An other construction is a self-standing construction that belongs to theme <i>Buildings</i> and that is not a Building.
Description:	NOTE 1: the main difference between a building and an other construction is the fact that an other construction does not need to be enclosed. NOTE 2: the other constructions to be considered under scope of theme <i>Buildings</i> are the constructions that are not present in another INSPIRE theme and that are necessary for environmental use cases, such as the ones considered in this data specification. EXAMPLES: bridge, acoustic fence, city wall.
Stereotypes:	«featureType»
Attribute: otherConstructionNature	
Name:	Other construction nature
Value type:	OtherConstructionNatureValue
Definition:	A description of the construction that represents its intended nature or current function and which differentiates it from that of a Building.
Multiplicity:	1
5.6.2.1.4. <i>BuildingAndBuildingUnitInfo</i>	
BuildingAndBuildingUnitInfo (abstract)	
Name:	Building and building unit info
Definition:	Abstract spatial object type grouping the additional properties that are common to Building , Building Part and BuildingUnit.
Description:	NOTE 1: The additional properties are those that are not already included in the base application schema. NOTE 2: These additional properties concern mainly the official information that may be attached to buildings / building parts or to building units.
Stereotypes:	«featureType»
Attribute: connectionToElectricity	
Name:	Connection to electricity
Value type:	Boolean
Definition:	An indication if the building or building part or building unit is connected or not to the public electricity network.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: connectionToGas	
Name:	Connection to gas
Value type:	Boolean
Definition:	An indication if the building or building part or building unit is connected or not to the public gas network.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: connectionToSewage	
Name:	Connection to sewage
Value type:	Boolean
Definition:	An indication if the building or building part or building unit is connected or not to the public sewage network.
Multiplicity:	0..1
Stereotypes:	«voidable»

BuildingAndBuildingUnitInfo (abstract)	
Attribute: connectionToWater	
Name:	Connection to water
Value type:	Boolean
Definition:	An indication if the building or building part or building unit is connected or not to the public water network.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: document	
Name:	Document
Value type:	Document
Definition:	Any document providing information about the building or building part or building unit.
Description:	EXAMPLES: the building permit, a photo of facade or inner yard, a sketch of interior, the building code, the energy performance assessment, an emergency plan
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: energyPerformance	
Name:	Energy performance
Value type:	EnergyPerformance
Definition:	The energy performance of the building or building part or building unit .
Description:	NOTE: The energy performance is required by the Energy Performance of Building Directive for the new buildings being rent or sold.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: heatingSource	
Name:	Heating source
Value type:	HeatingSourceValue
Definition:	The source of energy used for the heating
Description:	EXAMPLES: electricity, natural gas
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: heatingSystem	
Name:	Heating system
Value type:	HeatingSystemValue
Definition:	The system of heating
Description:	EXAMPLES : stove, central heating, heat pump
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: address	
Name:	Address
Value type:	AddressRepresentation
Definition:	The address(es) of the building or building part or building unit.
Description:	This attribute provides the current address(es) of the building or building component in the structured data type defined in theme Address.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: officialArea	

BuildingAndBuildingUnitInfo (abstract)	
Name:	Official area
Value type:	OfficialArea
Definition:	The area of the building or building part or building unit as registered in an official information system
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: officialValue	
Name:	Official value
Value type:	OfficialValue
Definition:	The value of the building or building part or building unit as registered in official information system
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: cadastralParcel	
Name:	Cadastral parcel
Value type:	CadastralParcel
Definition:	The cadastral parcel(s) to which the building or building part or building unit is officially related.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: address	
Name:	Address
Value type:	AddressRepresentation
Definition:	The address(es) of the building or building part or building unit.
Multiplicity:	0..*
Stereotypes:	«voidable»

5.6.2.1.5. *BuildingInfo*

BuildingInfo (abstract)	
Name:	Building info
Subtype of:	BuildingAndBuildingUnitInfo
Definition:	Abstract spatial object type grouping the additional specific properties of Building and Building Part.
Description:	NOTE 1: The additional properties are those that are not already included in the base application schema. NOTE 2: The specific properties are the properties that apply to Building and BuildingPart without applying to BuildingUnit.
Stereotypes:	«featureType»
Attribute: heightBelowGround	
Name:	Height below ground
Value type:	Length
Definition:	Height below ground of the building or building part.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: numberOffFloorsBelowGround	
Name:	Number of floors below ground
Value type:	Integer
Definition:	The number of floors below ground of the building or building part.
Description:	EXAMPLES: includes cellars, underground carparks ...

BuildingInfo (abstract)	
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: floorDistribution	
Name:	Floor distribution
Value type:	FloorRange
Definition:	The range(s) of floors of the building or building part.
Description:	EXAMPLE: [0,5] for a 6 floors building located on ground.
Multiplicity:	1..*
Stereotypes:	«voidable»
Attribute: floorDescription	
Name:	Floor description
Value type:	FloorDescription
Definition:	The description of a given range of building floors.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: roofType	
Name:	Roof type
Value type:	RoofTypeValue
Definition:	The shape of the roof.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: materialOfFacade	
Name:	Material of facade
Value type:	MaterialOfFacadeValue
Definition:	Material(s) of the building or building part facade.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: materialOfRoof	
Name:	Material of roof
Value type:	MaterialOfRoofValue
Definition:	Material(s) of the building or building part roof.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: materialOfStructure	
Name:	Material of structure
Value type:	MaterialOfStructureValue
Definition:	Material(s) of the building structure.
Description:	NOTE: generally, the building structure consists of the supporting walls or columns.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: buildingUnit	
Name:	Building unit
Value type:	AbstractBuildingUnit
Definition:	The building unit(s) belonging to the building or building part.
Multiplicity:	0..*
Stereotypes:	«voidable»

BuildingInfo (abstract)	
Association role: installation	
Value type:	AbstractInstallation
Definition:	The installation(s) serving the building or building part.
Multiplicity:	0..*
Stereotypes:	«voidable»

5.6.2.2. Data types

5.6.2.2.1. Document

Document	
Name:	Document
Definition:	This data types provides the address where the document may be found and a minimum set of metadata elements considered as necessary to exploit the document.
Stereotypes:	«dataType»
Attribute: date	
Name:	date
Value type:	DateTime
Definition:	Date of validity of the document.
Description:	EXAMPLES: the date the photo was taken, the date the sketch was done or approved, the date the building permit was accepted.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: documentDescription	
Name:	documentDescription
Value type:	PT_FreeText
Definition:	A short text providing overview of the document content. May be just title of the document.
Description:	EXAMPLES: "photo of inner yard", "sketch of third floor"
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: documentLink	
Name:	documentLink
Value type:	URI
Definition:	The Universal Resource Identifier of the document.
Description:	The Internet address where the document may be found.
Multiplicity:	1
Attribute: sourceStatus	
Name:	sourceStatus
Value type:	SourceStatusValue
Definition:	The status of the document, i.e. this attribute indicates if the document comes from official source or not.
Multiplicity:	1
Stereotypes:	«voidable»

5.6.2.2.2. EnergyPerformance

EnergyPerformance	
Name:	Energy performance
Definition:	This data type describes the energy performance of the building or building unit.
Stereotypes:	«dataType»

EnergyPerformance	
Attribute: energyPerformanceValue	
Name:	energyPerformanceValue
Value type:	EnergyPerformanceValue
Definition:	The class of energy performance of the building or building unit.
Multiplicity:	1
Attribute: dateOfAssessment	
Name:	dateOfAssessment
Value type:	DateTime
Definition:	The date when the energy performance of the building or building unit was assessed.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: assessmentMethod	
Name:	assessmentMethod
Value type:	DocumentCitation
Definition:	The reference to the document describing the assessment method of energy performance.
Multiplicity:	1
Stereotypes:	«voidable»

5.6.2.2.3. FloorDescription

FloorDescription	
Name:	Floor description
Definition:	This data type gathers the main characteristics of a floor (or range of floors) of a building.
Description:	The common characteristics are the ones coming from the use cases considered by this data specification.
Stereotypes:	«dataType»
Attribute: areaOfOpenings	
Name:	areaOfOpenings
Value type:	Area
Definition:	The area of openings (doors, windows, open space) on the facade of the building, related to this given floor
Description:	NOTE : the area of openings helps to assess the vulnerability of buildings to earthquakes.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: currentUse	
Name:	currentUse
Value type:	CurrentUseValue
Definition:	The current use(s) of the floor.
Multiplicity:	1..*
Stereotypes:	«voidable»
Attribute: document	
Name:	document
Value type:	Document
Definition:	Any document providing information about the floor.
Description:	EXAMPLE : A sketch of the floor, emergency plan of the floor.
Multiplicity:	0..*

FloorDescription	
Stereotypes:	«voidable»
Attribute: floorArea	
Name:	floorArea
Value type:	Area
Definition:	The ground area of the floor.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: floorRange	
Name:	floorRange
Value type:	FloorRange
Definition:	The range of floors having common characteristics.
Description:	NOTE: Many buildings may have ground floor with specific characteristics and upper floors looking like one another. EXAMPLE 1: Typically, the ground floor may be used for shops and the upper floors for offices or dwellings. The opening distribution is also often different on ground floor (with entrance doors, arcades, ...) and in upper floors (with only windows on the facade).
Multiplicity:	1..*
Attribute: height	
Name:	height
Value type:	Length
Definition:	The height of the floor.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: numberOfDwellings	
Name:	numberOfDwellings
Value type:	Integer
Definition:	The number of dwellings of the floor.
Multiplicity:	1
Stereotypes:	«voidable»

5.6.2.2.4. FloorRange

FloorRange	
Name:	FloorRange
Definition:	The identification of a floor range by its lowest and highest floor.
Description:	NOTE 1: The ground floor should be considered as floor n°0. NOTE 2: If the floor range includes only one floor, the lowest and highest floor will be equal, e.g. [0,0] to identify the ground floor. NOTE 3: In case of a building with several building parts, the same floor should be used as reference floor, i.e. as floor n° 0.
Stereotypes:	«dataType»
Attribute: lowestFloor	
Name:	lowestFloor
Value type:	Real
Definition:	The lowest floor in the floor range.
Description:	NOTE: lowestFloor is defined as float to deal with half floors that are used by some data producers (e.g. for mezzanines). Only numbers such as .. -2, -1, 0, 1, 2, ... or ..., -1.5, -0.5, 0.5, 1.5, 2.5, ... should be used to define lowest floor.
Multiplicity:	1

FloorRange	
Attribute: highestFloor	
Name:	highestFloor
Value type:	Real
Definition:	The highest floor in the floor range.
Description:	NOTE : HighestFloor is defined as float to deal with half floors that are used by some data producers (e.g. for mezzanines). Only numbers such as .. -2, -1, 0, 1, 2, ... or ..., -1,5, -0.5, 0.5, 1.5, 2.5, ... should be used to define highest floor.
Multiplicity:	1

5.6.2.2.5. OfficialArea

OfficialArea	
Name:	Official area
Definition:	This data types includes the official area of the building, building part or building unit and information about the exact meaning of this area.
Stereotypes:	«dataType»
Attribute: officialAreaReference	
Name:	officialAreaReference
Value type:	OfficialAreaReferenceValue
Definition:	The type of the official area.
Description:	The type of official area may be described either by using the values provided by the CLGE measurement code for the floor area of buildings (which values are provided by the CLGE_OfficialAreaReferenceValue) or by using another standard (which values are provided by the empty code list OtherStandard OfficialAreaReferenceValue, this code list having to be defined at Member State level).
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: value	
Name:	Value
Value type:	Area
Definition:	The value of the official area
Multiplicity:	1
Attribute: heightParameter	
Name:	heightParameter
Value type:	Length
Definition:	The height parameter used to differentiate internal primary area of internal other area, if the official area is referenced using the CLGE measurement code for the floor area of buildings
Description:	NOTE: According to CLGE code, the height parameter has a default value fixed to 2.10 m but may be changed in order to fit with national regulation.
Multiplicity:	0..1
Stereotypes:	«voidable»
Constraint: valueUoM	
Natural language:	Unit of value must be square meter.
OCL:	inv: self.value.uom.uomSymbol='m2'

5.6.2.2.6. OfficialValue

OfficialValue	
Name:	Official value

OfficialValue	
Definition:	Data type grouping the information about the official value itself and the metadata attached to it.
Description:	The official value may be provided either directly by a value and its currency , or e.g. in case of privacy issues, by an external reference to another information system. This official value generally aims to assess the market price (valueReference) of the building (or building unit) or a given percentage of this valueReference at a valuationDate.
Stereotypes:	«dataType»
Attribute: currency	
Name:	currency
Value type:	CurrencyValue
Definition:	The currency in which the official value is provided.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: value	
Name:	value
Value type:	Integer
Definition:	The official value of the building or building unit.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: valuationDate	
Name:	valuationDate
Value type:	DateTime
Definition:	The date corresponding to the assessed market price.
Description:	EXAMPLE: The official value aims to assess the market price as it was in January 2012.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: officialValueReference	
Name:	officialValueReference
Value type:	OfficialValueReferenceValue
Definition:	The reference market price that the official value aims to assess.
Description:	EXAMPLE: rental income
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: referencePercentage	
Name:	referencePercentage
Value type:	Integer
Definition:	The percentage of the market price that the official value aims to assess.
Description:	The official value aims to assess 50% of market price.
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: informationSystemName	
Name:	informationSystemName
Value type:	PT_FreeText
Definition:	The name of an external information system where the official value may be found.
Description:	It will be possible to find the official value of the building, building part or building

OfficialValue	
	unit, using the external reference of the spatial object related to the given information system.
Multiplicity:	1
Stereotypes:	«voidable»
Constraint: either value and currency or informationSystemName shall be provided	
Natural language:	Either value and currency or informationSystemName shall be provided.
OCL:	inv: (value->notEmpty() and currency->notEmpty()) or (informationSystemName->notEmpty())
Constraint: informationSystemName shall be present in one external reference	
Natural language:	The informationSystemName shall be present in one of the external references of the spatial object.
OCL:	

5.6.2.3. Code lists

5.6.2.3.1. CLGE_OfficialAreaReferenceValue

CLGE_OfficialAreaReferenceValue	
Name:	CLGE_OfficialAreaReferenceValue
Definition:	List of values for the reference of official area, as defined in the CLGE measurement code for the floor area of buildings. SOURCE: http://www.eureal.eu/
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/OfficialAreaReferenceValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.2. CurrencyValue

CurrencyValue	
Name:	CurrencyValue
Definition:	Code list for possible values of attribute currency
Description:	NOTE 1: include currencies from all European countries, including that are not Member States of European Union. SOURCE: values are extracted from ISO 4217 standard. NOTE 2: this code list may be of interest not only for INSPIRE but also for other European applications and regulations ; so, in future, this code list might/should be managed outside INSPIRE.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/CurrencyValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.3. EnergyPerformanceValue

EnergyPerformanceValue	
Name:	EnergyPerformanceValue
Definition:	Code list for possible values of energy performance of a building or building part or building unit.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/EnergyPerformanceValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.4. HeatingSourceValue

HeatingSourceValue

Name:	HeatingSourceValue
Definition:	Code list for the possible values of the heating source of a building, building part or building unit.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/HeatingSourceValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.5. HeatingSystemValue

HeatingSystemValue

Name:	HeatingSystemValue
Definition:	Code list giving the possible values for the heating system of a building, building part or building unit.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/HeatingSystemValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.6. InstallationNatureValue

InstallationNatureValue

Name:	InstallationNatureValue
Definition:	Code list giving the possible values of an installation nature.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/InstallationNatureValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.7. MaterialOfFacadeValue

MaterialOfFacadeValue

Name:	MaterialOfFacadeValue
Definition:	Code list for the possible values of MaterialOfFacade
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/MaterialOfFacadeValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.8. MaterialOfRoofValue

MaterialOfRoofValue

Name:	MaterialOfRoofValue
Definition:	Code list for possible values of attribute MaterialOfRoof
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/MaterialOfRoofValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.9. MaterialOfStructureValue

MaterialOfStructureValue

Name:	MaterialOfStructureValue
-------	--------------------------

MaterialOfStructureValue

Definition:	Code list for possible values of attribute MaterialOfStructure.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/MaterialOfStructureValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.10. OfficialAreaReferenceValue

OfficialAreaReferenceValue

Name:	OfficialAreaReferenceValue
Definition:	The list of possible values for the reference of the official area.
Description:	The type of official area may be described either by using the values provided by the CLGE measurement code for the floor area of buildings (which values are provided by the CLGE_OfficialAreaReferenceValue) or by using another standard (which values are provided by the empty code list OtherStandard OfficialAreaReferenceValue, this code list having to be defined at Member State level).
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/OfficialAreaReferenceValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.11. OfficialValueReferenceValue

OfficialValueReferenceValue

Name:	OfficialValueReferenceValue
Definition:	The list of possible values for referencing the official value of a building, building part or building unit.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/OfficialValueReferenceValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.12. OtherStandardOfficialAreaReferenceValue

OtherStandardOfficialAreaReferenceValue

Name:	Other standard official area reference value
Definition:	Reference to a standard for official area code list.
Extensibility:	open
Identifier:	
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.13. RoofTypeValue

RoofTypeValue

Name:	RoofTypeValue
Definition:	Code list for the possible values of attribute roofType.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/RoofTypeValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.14. SourceStatusValue

SourceStatusValue

Name:	SourceStatusValue
Definition:	Code list for possible values of attribute sourceStatus (of Document).
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/SourceStatusValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.3.15. OtherConstructionNatureValue

OtherConstructionNatureValue

Name:	OtherConstructionNatureValue
Definition:	Code list for the attribute other construction nature.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/OtherConstructionNatureValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.6.2.4. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

5.6.2.4.1. AbstractConstruction

AbstractConstruction (abstract)

Package:	BuildingsBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the semantic properties of buildings, building parts and of some optional spatial object types that may be added in order to provide more information about the theme <i>Buildings</i> .
Description:	The optional spatial object types that may be added to core profiles are described in the extended profiles. The ones inheriting from the attributes of AbstractConstruction are Installation and OtherConstruction.

5.6.2.4.2. AddressRepresentation

AddressRepresentation

Package:	Addresses
Reference:	INSPIRE Data specification on Addresses [DS-D2.8.I.5]
Definition:	Representation of an address spatial object for use in external application schemas that need to include the basic, address information in a readable way.
Description:	NOTE 1 The data type includes the all necessary readable address components as well as the address locator(s), which allows the identification of the address spatial objects, e.g., country, region, municipality, address area, post code, street name and address number. It also includes an optional reference to the full address spatial object. NOTE 2 The datatype could be used in application schemas that wish to include address information e.g. in a dataset that registers buildings or properties.

5.6.2.4.3. Area

Area

Package:	Units of Measure
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.6.2.4.4. *Boolean*

Boolean

Package:	Truth
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.6.2.4.5. *CadastralParcel*

CadastralParcel

Package:	CadastralParcels
Reference:	INSPIRE Data specification on Cadastral Parcels [DS-D2.8.I.6]
Definition:	Areas defined by cadastral registers or equivalent.
Description:	SOURCE [INSPIRE Directive:2007].

NOTE As much as possible, in the INSPIRE context, cadastral parcels should be forming a partition of national territory. Cadastral parcel should be considered as a single area of Earth surface (land and/or water), under homogeneous real property rights and unique ownership, real property rights and ownership being defined by national law (adapted from UN ECE 2004 and WG-CPI, 2006). By unique ownership is meant that the ownership is held by one or several joint owners for the whole parcel.

5.6.2.4.6. *CurrentUseValue*

CurrentUseValue

Package:	BuildingsBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	List of possible values indicating the current use.
Description:	SOURCE: This code list is partly based on and adapted from the Eurostat classification of types of constructions (for the classification of residential buildings). NOTE: the values of this code list apply to buildings or building components where building components may be a building part (in core profiles) or a building unit (in extended profiles)

5.6.2.4.7. *DateTime*

DateTime

Package:	Date and Time
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.6.2.4.8. *DocumentCitation*

DocumentCitation

Package:	Base Types 2
Reference:	INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5]
Definition:	Citation for the purposes of unambiguously referencing a document.

5.6.2.4.9. *ExternalReference*

ExternalReference

Package:	BuildingsBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Reference to an external information system containing any piece of information related to the spatial object.

5.6.2.4.10. *Identifier*

Identifier

Package:	Base Types
Reference:	INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5]
Definition:	External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.

Identifier	
Description:	<p>NOTE1 External object identifiers are distinct from thematic object identifiers.</p> <p>NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object.</p> <p>NOTE 3 The unique identifier will not change during the life-time of a spatial object.</p>

5.6.2.4.11. *Integer*

Integer	
Package:	Numerics
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.6.2.4.12. *Length*

Length	
Package:	Units of Measure
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.6.2.4.13. *PT_FreeText*

PT_FreeText	
Package:	Cultural and linguistic adaptability
Reference:	Geographic information -- Metadata -- XML schema implementation [ISO/TS 19139:2007]

5.6.2.4.14. *Real*

Real	
Package:	Numerics
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.6.2.4.15. *URI*

URI	
Package:	basicTypes
Reference:	Geographic information -- Geography Markup Language (GML) [ISO 19136:2007]

5.7 Application schema BuildingsExtended2D

5.7.1 Description

5.7.1.1. Narrative description

<Buildings Extended 2D> application schema is an illustrative profile. It aims to be a recommendation for data providers willing to give more information than the basic one included in core 2D profile. This extended profile may be used as a whole or only partly, i.e. only a selection of proposed feature types and attributes may be added. More detailed information is provided in annex F.

Extended2D profile is both:

- an extension of <Buildings 2D> profile, i.e. it includes buildings and building parts with their 2D geometric representation (defined in <Base 2D>) and the basic core properties inherited from <Buildings Base>
- an extension of <Building BaseExtended>, i.e. it includes its additional feature types (other constructions, installations and building units) and its additional properties, such as official information and more detailed physical description.

The application schema <Buildings extended2D> does not define any other attribute for buildings and building parts.

- Feature types OtherConstruction and Installation get the attribute about “nature” defined in the <Buildings Base extended> application schema and the core attributes inherited from AbstractConstruction in <Buildings Base> application schema.

In addition, <Buildings Extended2D> defines their 2D geometric representation as a generic GM_Object.

- Feature types BuildingUnit inherits only the attributes defined directly in <Buildings Base Extended>, namely the specific attributes of AbstractBuildingUnit (e.g. the mandatory external reference) and the attributes common with buildings and building parts defined in BuildingAndBuildingUnitInfo.

In addition, <Buildings Extended2D> defines their 2D geometric representation as a generic GM_Object. Note that, in opposition to the other feature types of theme Buildings, this geometry is voidable

5.7.1.2. UML Overview

See previous Figure 53.

5.7.2 Feature catalogue

Feature catalogue metadata

Application Schema	INSPIRE Application Schema BuildingsExtended2D
Version number	3.0

Types defined in the feature catalogue

Type	Package	Stereotypes
<i>Building</i>	BuildingsExtended2D	«featureType»
<i>BuildingPart</i>	BuildingsExtended2D	«featureType»
<i>BuildingUnit</i>	BuildingsExtended2D	«featureType»
<i>Installation</i>	BuildingsExtended2D	«featureType»
<i>OtherConstruction</i>	BuildingsExtended2D	«featureType»

5.7.2.1. Spatial object types

5.7.2.1.1. BuildingUnit

BuildingUnit	
Name:	Building unit
Subtype of:	AbstractBuildingUnit
Definition:	A BuildingUnit is a subdivision of Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc.
Description:	Building units are spatial objects aimed at subdividing buildings and/or building parts into smaller parts that are treated as separate entities in daily life. A building unit is homogeneous, regarding management aspects. EXAMPLES: It may be e.g. an apartment in a condominium, a terraced house, or a shop inside a shopping arcade. NOTE 1: According to national regulations, a building unit may be a flat, a cellar, a garage or set of a flat, a cellar and a garage. NOTE 2: According to national regulation, a building that is one entity for daily life (typically, a single family house) may be considered as a Building composed of one BuildingUnit or as a Building composed of zero BuildingUnit.
Stereotypes:	«featureType»

BuildingUnit	
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	GM_Primitive
Definition:	The 2D or 2,5 D geometric representation of the building unit.
Description:	EXAMPLE: the building unit may be represented by its floor surface or by a simple point.
Multiplicity:	1
Stereotypes:	«voidable»

5.7.2.1.2. Installation

Installation	
Name:	Installation
Subtype of:	AbstractInstallation
Definition:	An external construction (of small size) or an external device serving the building or building part.
Description:	EXAMPLES: stairway, solar panel, external lift
Stereotypes:	«featureType»
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	GM_Primitive
Definition:	2D or 2,5 D geometric representation of the other construction.
Multiplicity:	1

5.7.2.1.3. OtherConstruction

OtherConstruction	
Name:	Other construction
Subtype of:	AbstractOtherConstruction
Definition:	An OtherConstruction is a self-standing construction that belongs to theme Buildings and that is not a Building.
Description:	NOTE 1: the main difference between a building and an other construction is the fact that an other construction does not need to be enclosed. NOTE 2: the other constructions to be considered under scope of theme buildings are the constructions that are not present in another INSPIRE theme and that are necessary for environmental use cases, such as the ones considered in this data specification. EXAMPLES: bridge, acoustic fence, city wall.
Stereotypes:	«featureType»
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	GM_Primitive
Definition:	Geometric representation of the building.
Multiplicity:	1

5.7.2.1.4. Building

Building	
Name:	Building
Subtype of:	BuildingInfoBuilding
Definition:	A Building is an enclosed construction above and/or underground, used or intended for the shelter of humans, animals or things or for the production of economic goods. A building refers to any structure permanently constructed or erected on its site.
Stereotypes:	«featureType»

5.7.2.1.5. *BuildingPart*

BuildingPart

Name:	Building part
Subtype of:	BuildingInfoBuildingPart
Definition:	A BuildingPart is a sub-division of a Building that might be considered itself as a building.
Description:	NOTE 1: A BuildingPart is homogeneous related to its physical, functional or temporal aspects. NOTE 2: Building and BuildingPart share the same set of properties. EXAMPLE: A Building may be composed of two BuildingParts having different heights above ground.
Stereotypes:	«featureType»

5.7.2.2. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

5.7.2.2.1. *AbstractBuildingUnit*

AbstractBuildingUnit (abstract)

Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the semantic properties of building units. A BuildingUnit is a subdivision of Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc.
Description:	Building units are spatial objects aimed at subdividing buildings and/or building parts into smaller parts that are treated as separate entities in daily life. A building unit is homogeneous, regarding management aspects. EXAMPLES: It may be e.g. an apartment in a condominium, a terraced house, or a shop inside a shopping arcade. NOTE 1: According to national regulations, a building unit may be a flat, a cellar, a garage or set of a flat, a cellar and a garage. NOTE 2: According to national regulation, a building that is one entity for daily life (typically, a single family house) may be considered as a Building composed of one BuildingUnit or as a Building composed of zero BuildingUnit.

5.7.2.2.2. *AbstractInstallation*

AbstractInstallation (abstract)

Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the semantic properties of installations. An external construction (of small size) or an external device serving the building or building part.
Description:	EXAMPLES: stairway, solar panel, external lift

5.7.2.2.3. *AbstractOtherConstruction*

AbstractOtherConstruction (abstract)

Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the semantic properties of other constructions. An other construction is a self-standing construction that belongs to theme <i>Buildings</i> and that is not a Building.
Description:	NOTE 1: the main difference between a building and an other construction is the

AbstractOtherConstruction (abstract)

fact that an other construction does not need to be enclosed.
NOTE 2: the other constructions to be considered under scope of theme *Buildings* are the constructions that are not present in another INSPIRE theme and that are necessary for environmental use cases, such as the ones considered in this data specification.
EXAMPLES: bridge, acoustic fence, city wall.

5.7.2.2.4. BuildingInfo

BuildingInfo (abstract)

Package: BuildingsExtendedBase
Reference: INSPIRE Data specification on *Buildings* [DS-D2.8.III.2]
Definition: Abstract spatial object type grouping the additional specific properties of Building and Building Part.
Description: NOTE 1: The additional properties are those that are not already included in the base application schema.
NOTE 2: The specific properties are the properties that apply to Building and BuildingPart without applying to BuildingUnit.

5.7.2.2.5. GM_Primitive

GM_Primitive (abstract)

Package: Geometric primitive
Reference: Geographic information -- Spatial schema [ISO 19107:2003]

5.8 Application schema BuildingsExtended3D

5.8.1 Description

5.8.1.1. Narrative description

<Buildings Extended 3D> application schema is an illustrative profile. It aims to be a recommendation for data providers willing to give more information than the basic one included in core 3D profile.

This extended profile may be used as a whole or only partly, i.e. only a selection of proposed feature types and attributes may be added. More detailed information is provided in annex F.

<Buildings Extended3D> profile is both:

- an extension of <Buildings 3D> application schema, i.e. it includes buildings and building parts with their 2D geometric representation (defined in <Base 3D>) and the basic core properties inherited from <Buildings Base>
- an extension of <Building BaseExtended>, i.e. it includes its additional feature types (other constructions, installations and building units) and its additional properties, such as official information and more detailed physical description.

In addition, <Buildings Extended3D> offers a more detailed description of buildings and building parts by defining their physical components, such as roof, walls, ground, openings, room and by describing mechanisms to attach a texture to these components.

5.8.1.1.1. *Main feature types*

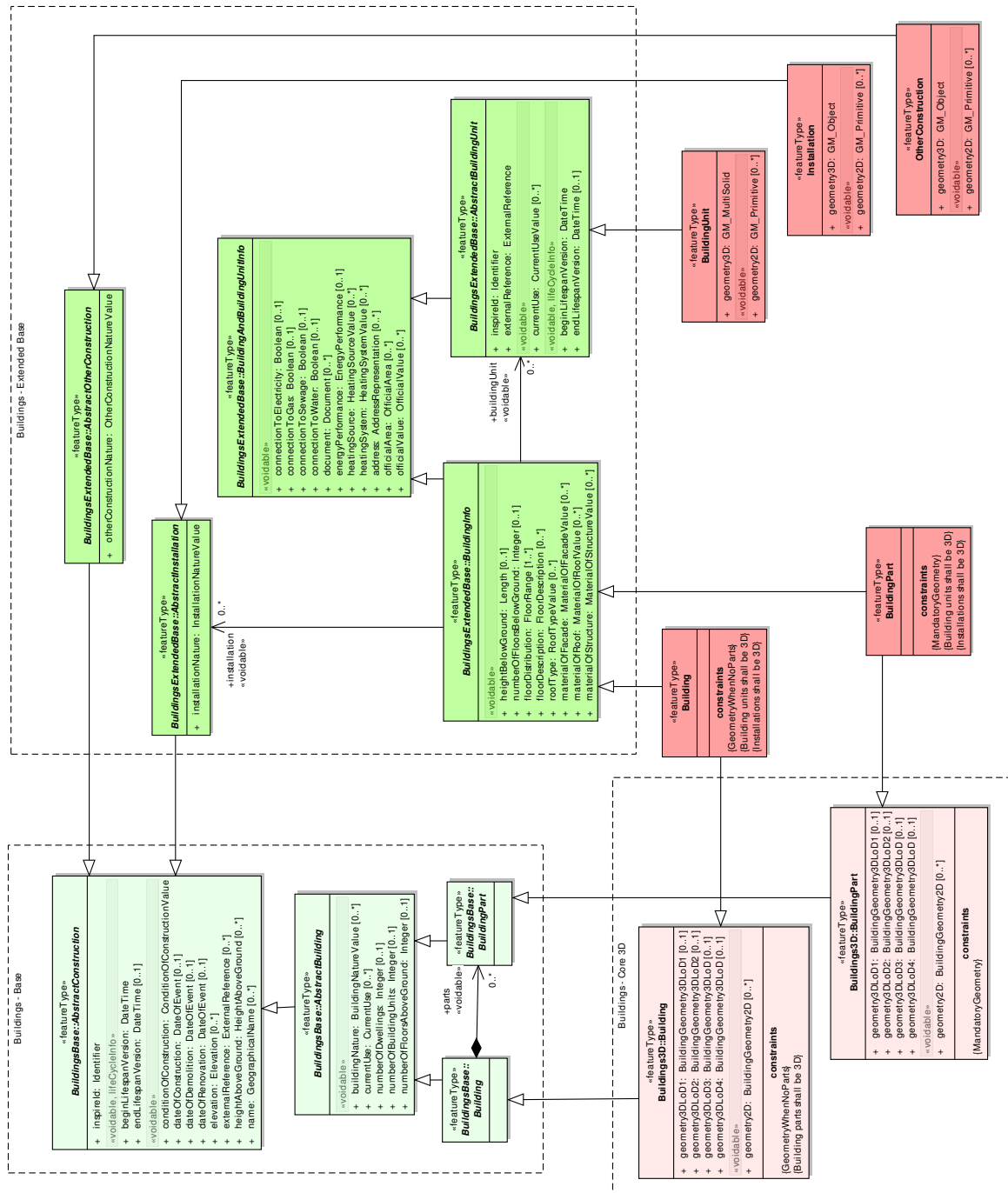


Figure 54: The main feature types of <Buildings Extended3D>

In the above figure, the feature types defined by <Buildings extended3D> are displayed in pink.

- Feature types Building and BuildingPart inherit both from
 - <Buildings 3D> : buildings and building parts get the 3D geometric representation defined in this application schema and the core semantics inherited from <Buildings Base> application schema
 - <Buildings Base extended>: building and building parts get the additional properties defined in this application schema, namely the official information coming from BuildingAndBuildingUnitInfo and the more detailed topographic description coming from BuildingInfo.

The application schema <Buildings extended3D> does not define any other attribute for buildings and building parts.

- Feature types OtherConstruction and Installation get the attribute about “nature” defined in the <Buildings Base extended> application schema and the core attributes inherited from AbstractConstruction in <Buildings Base> application schema.

In addition, <Buildings Extended2D> defines their 3D geometric representation as a generic GM_Object (that shall obviously be given with 3D coordinates).

- Feature types BuildingUnit inherits only the attributes defined directly in <Buildings Base Extended>, namely the specific attributes of AbstractBuildingUnit (e.g. the mandatory external reference) and the attributes common with buildings and building parts defined in BuildingAndBuildingUnitInfo.

In addition, <Buildings Extended3D> defines their 3D geometric representation as a GM_MultiSolid.

5.8.1.1.2. The 3D building model

As in <Buildings 2D>, the buildings and building parts may be represented by a solid or a multi-surface, according to the various LoDs of City GML.

Moreover, the components of the building may also be semantically described by specific feature types:

- in LoD2 : boundary surfaces(e.g. walls and roof) and installations
- in LoD3 : openings (doors and windows) are added
- in LoD4 : the interior of building (building units, rooms, internal installations) are added.

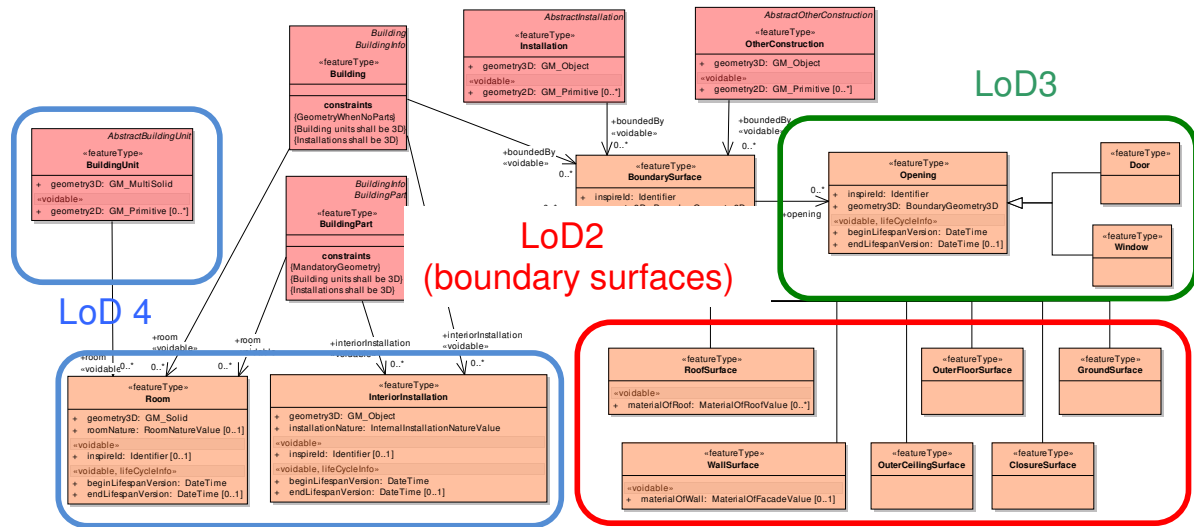


Figure 55: Modeling of LoD2, LoD3 and LoD4

In the Figure 55, the main feature types (Building, BuildingPart, BuildingUnit, Installation, OtherConstruction) are represented in pink, without their attributes. These attributes have been described in previous paragraph and figure. The new feature types are represented in orange.

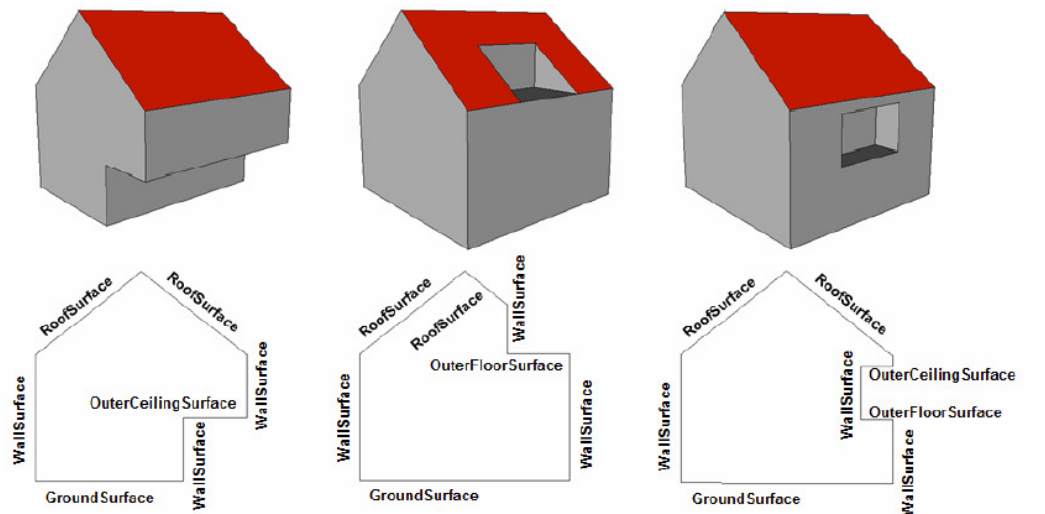


Figure 56: Examples of use of OuterFloorSurface and OuterCeilingSurface

LoD 4 relates to description of building interior. In CityGML, it is limited to the representation of Rooms and InteriorInstallations; in INSPIRE model, the representation of BuildingUnits has been added. Feature types BuildingUnit and Room may be represented separately or together; in last case, the BuildingUnit will be composed of Rooms.

5.8.1.1.3. Geometry of 3D feature types

The geometry of 3D feature types has to be provided using one of these 5 types:

- **BuildingGeometry3D**: it is the data type defined in core 3D profile. It is used to represent the 2 core feature types : buildings and building parts.
- **GM_Solid or GM_MultiSolid**: it is the simple geometry primitive to represent the volumetric features related to of the building interior, i.e. rooms and building units. Note that these feature types have to be represented only in LoD4. The GM_Solid has to be used for rooms and the GM_MultiSolid for the building units.
- **BoundaryGeometry3D** : this data type has to be used to represent the objects that are surfaces, i.e. wall surfaces, roof surfaces, closure surfaces, ground surfaces, outer ceiling surfaces, outer floor surface and openings (doors and windows). The boundary surface may be represented at different levels of detail, namely LoD2, LoD3 and LoD4. It is recommended to provide the accuracy of this geometric representation, both in its horizontal and vertical dimensions (see figure below).
- **GM_Object**: this generic geometric primitive has to be used to represent the objects whose shape may be a volume, a surface or a line. This data type is used for internal and external installations and for other constructions. For instance:
 - An antenna may be represented by a vertical line
 - Solar panel may be represented by a surface
 - Dormer may be represented by a volume / solid.

NOTE: In <Buildings Extended 3D> spatial objects may be solid, surfaces, lines or even points but their geometry has to be given with 3 coordinates.

<div>«dataType»</div> <div>BoundaryGeometry3D</div>
<div>+ LoD2MultiSurface: GM_MultiSurface [0..1]</div> <div>+ LoD3MultiSurface: GM_MultiSurface [0..1]</div> <div>+ LoD4MultiSurface: GM_MultiSurface [0..1]</div> <div>«voidable»</div> <div>+ horizontalGeometryEstimatedAccuracyLoD2: Length [0..1]</div> <div>+ horizontalGeometryEstimatedAccuracyLoD3: Length [0..1]</div> <div>+ horizontalGeometryEstimatedAccuracyLoD4: Length [0..1]</div> <div>+ verticalGeometryEstimatedAccuracyLoD2: Length [0..1]</div> <div>+ verticalGeometryEstimatedAccuracyLoD3: Length [0..1]</div> <div>+ verticalGeometryEstimatedAccuracyLoD4: Length [0..1]</div>
<div>constraints</div> <div>{atLeastOneMandatoryGeometry}</div>

Figure 57: The data type BoundaryGeometry3D

5.8.1.1.4. The appearance model

The INSIRE appearance model is based on City GML one. See more details about how a texture may be attached to the main features of the building model in annex D.

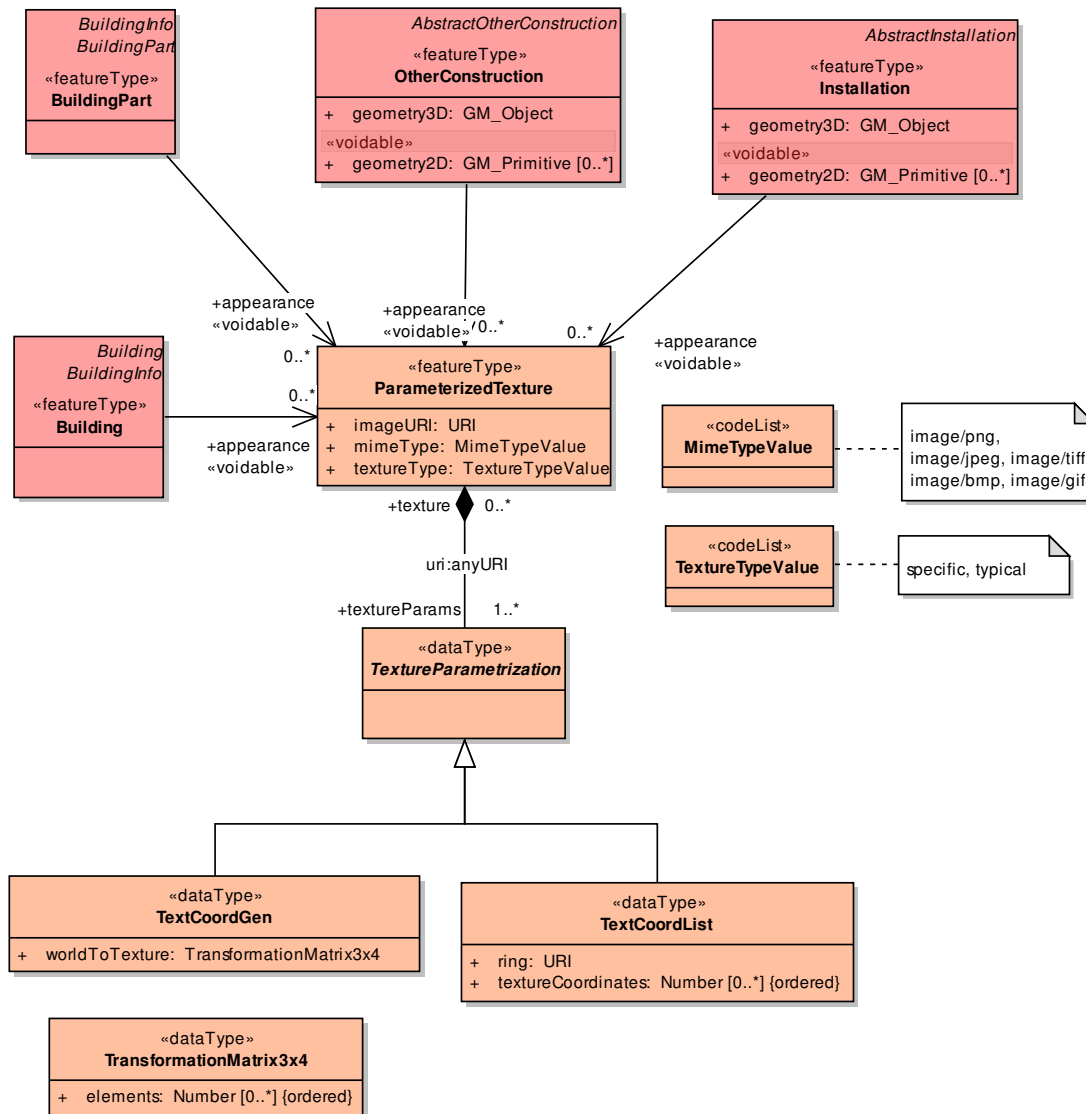


Figure 58: The appearance part (simplification of CityGML appearance model)

5.8.1.2. UML Overview

See previous figures.

5.8.1.3. Identifier management

NOTE: For the detailed feature types of <Buildings Extended3D> application schema (such as roofs, walls, openings, rooms), the attribute inspireId is voidable whereas it is mandatory for the main feature types (buildings, buildings parts, installations, building units, other constructions).

5.8.1.4. Geometry representation

Art. 12(1) of Regulation 1089/2010 restricts the value domain of spatial properties to the Simple Feature spatial schema as defined in the *OpenGIS® Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture, version 1.2.1*, unless specified otherwise for a specific spatial data theme or type.

TG Requirement By way of derogation from article 12(1), the value domain of spatial properties used in the Buildings 3D package shall not be restricted to simple feature.

5.8.2 Feature catalogue

Feature catalogue metadata

Application Schema	INSPIRE Application Schema BuildingsExtended3D
Version number	3.0

Types defined in the feature catalogue

Type	Package	Stereotypes
<i>BoundaryGeometry3D</i>	BuildingsExtended3D	«dataType»
<i>BoundarySurface</i>	BuildingsExtended3D	«featureType»
<i>Building</i>	BuildingsExtended3D	«featureType»
<i>BuildingPart</i>	BuildingsExtended3D	«featureType»
<i>BuildingUnit</i>	BuildingsExtended3D	«featureType»
<i>ClosureSurface</i>	BuildingsExtended3D	«featureType»
<i>Door</i>	BuildingsExtended3D	«featureType»
<i>GroundSurface</i>	BuildingsExtended3D	«featureType»
<i>Installation</i>	BuildingsExtended3D	«featureType»
<i>InteriorInstallation</i>	BuildingsExtended3D	«featureType»
<i>InternalInstallationNatureValue</i>	BuildingsExtended3D	«codeList»
<i>MimeTypeValue</i>	BuildingsExtended3D	«codeList»
<i>Opening</i>	BuildingsExtended3D	«featureType»
<i>OtherConstruction</i>	BuildingsExtended3D	«featureType»
<i>OuterCeilingSurface</i>	BuildingsExtended3D	«featureType»
<i>OuterFloorSurface</i>	BuildingsExtended3D	«featureType»
<i>ParameterizedTexture</i>	BuildingsExtended3D	«featureType»
<i>RoofSurface</i>	BuildingsExtended3D	«featureType»
<i>Room</i>	BuildingsExtended3D	«featureType»
<i>RoomNatureValue</i>	BuildingsExtended3D	«codeList»
<i>TextCoordGen</i>	BuildingsExtended3D	«dataType»
<i>TextCoordList</i>	BuildingsExtended3D	«dataType»
<i>TextureParametrization</i>	BuildingsExtended3D	«dataType»
<i>TextureTypeValue</i>	BuildingsExtended3D	«codeList»
<i>TransformationMatrix3x4</i>	BuildingsExtended3D	«dataType»
<i>WallSurface</i>	BuildingsExtended3D	«featureType»
<i>Window</i>	BuildingsExtended3D	«featureType»

5.8.2.1. Spatial object types

5.8.2.1.1. BoundarySurface

BoundarySurface	
Name:	BoundarySurface

BoundarySurface	
Definition:	Spatial objects being part of the building exterior shell with a special function.
Description:	EXAMPLES: wall (WallSurface), roof (RoofSurface), ground plate (GroundSurface) or ClosureSurface.
Stereotypes:	«featureType»
Attribute: inspireId	
Name:	inspireId
Value type:	Identifier
Definition:	External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.
Description:	NOTE1 External object identifiers are distinct from thematic object identifiers. NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object. NOTE 3 The unique identifier will not change during the life-time of a spatial object.
Multiplicity:	1
Attribute: geometry3D	
Name:	geometry3D
Value type:	BoundaryGeometry3D
Definition:	The 3D geometric representation of the boundary (surface embedded in 3D).
Multiplicity:	1
Attribute: beginLifespanVersion	
Name:	beginLifespanVersion
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
Multiplicity:	1
Stereotypes:	«voidable,lifeCycleInfo»
Attribute: endLifespanVersion	
Name:	endLifespanVersion
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
Multiplicity:	0..1
Stereotypes:	«voidable,lifeCycleInfo»
Association role: opening	
Value type:	Opening
Definition:	The opening(s) being part of the boundary surface.
Multiplicity:	0..*

5.8.2.1.2. Building

Building	
Name:	Building
Subtype of:	BuildingInfoBuilding
Definition:	A Building is an enclosed construction above and/or underground, used or intended for the shelter of humans, animals or things or for the production of economic goods. A building refers to any structure permanently constructed or erected on its site.
Stereotypes:	«featureType»

Building	
Association role: interiorInstallation	
Value type:	InteriorInstallation
Definition:	The interior installation(s) serving the building.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: boundedBy	
Value type:	BoundarySurface
Definition:	The surface(s) bounding the building.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: appearance	
Value type:	ParameterizedTexture
Definition:	The texture(s) attached to the building and giving it its appearance.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: room	
Value type:	Room
Definition:	The room(s) composing the building.
Multiplicity:	0..*
Stereotypes:	«voidable»
Constraint: Building units shall be 3D	
Natural language:	The building units associated with the building shall be represented using the BuildingUnit type of the Buildings extended 3D package.
OCL:	inv: self.buildingUnit->oclIsKindOf(Buildings3D::BuildingUnit)
Constraint: GeometryWhenNoParts	
Natural language:	
OCL:	A Building without BuildingParts must provide at least one geometry3D. If BuildingParts are attached to the Building, the geometry3D is optional.
Constraint: Installations shall be 3D	
Natural language:	The installations associated with the building shall be represented using the Installationtype of the Buildings3D extended package.
OCL:	inv: self.installation->oclIsKindOf(Buildings3D::Installation)

5.8.2.1.3. BuildingPart

BuildingPart	
Name:	Building part
Subtype of:	BuildingInfoBuildingPart
Definition:	A BuildingPart is a sub-division of a Building that might be considered itself as a building.
Description:	<p>NOTE 1: A BuildingPart is homogeneous related to its physical, functional or temporal aspects</p> <p>NOTE2: Building and BuildingPart share the same set of properties.</p> <p>EXAMPLE: A building may be composed of two building parts having different heights above ground.</p>
Stereotypes:	«featureType»

BuildingPart	
Association role: interiorInstallation	
Value type:	InteriorInstallation
Definition:	The interior installation(s) serving the building part.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: appearance	
Value type:	ParameterizedTexture
Definition:	The texture(s) attached to the building part and giving it its appearance.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: boundedBy	
Value type:	BoundarySurface
Definition:	The surface(s) bounding the building part.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: room	
Value type:	Room
Definition:	The room(s) composing the building part.
Multiplicity:	0..*
Stereotypes:	«voidable»
Constraint: Building units shall be 3D	
Natural language:	The building units associated with the building shall be represented using the BuildingUnit type of the Buildings3D extended package.
OCL:	inv: self.buildingUnit->oclIsKindOf(Buildings3D::BuildingUnit)
Constraint: Installations shall be 3D	
Natural language:	The installations associated with the building shall be represented using the Installation type of the Buildings3D extended package.
OCL:	inv: self.installation->oclIsKindOf(Buildings3D::Installation)
Constraint: MandatoryGeometry	
Natural language:	
OCL:	A BuildingPart must provide at least one geometry3D.

5.8.2.1.4. *BuildingUnit*

BuildingUnit	
Name:	Building unit
Subtype of:	AbstractBuildingUnit
Definition:	A BuildingUnit is a subdivision of Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc.
Description:	Building units are spatial objects aimed at subdividing buildings and/or building parts into smaller parts that are treated as separate entities in daily life. A building unit is homogeneous, regarding management aspects. EXAMPLES: It may be e.g. an apartment in a condominium, a terraced house, or a shop inside a shopping arcade. NOTE 1: According to national regulations, a building unit may be a flat, a cellar, a garage or set of a flat, a cellar and a garage. NOTE 2: According to national regulation, a building that is one entity for daily

BuildingUnit	
Stereotypes:	life (typically, a single family house) may be considered as a Building composed of one BuildingUnit or as a Building composed of zero BuildingUnit. «featureType»
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	GM_Primitive
Definition:	2D or 2.5D geometric representation.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: geometry3D	
Name:	Geometry 3D
Value type:	GM_MultiSolid
Definition:	3D geometric representation.
Multiplicity:	1
Association role: room	
Value type:	Room
Definition:	The room(s) composing the building unit.
Multiplicity:	0..*
Stereotypes:	«voidable»

5.8.2.1.5. ClosureSurface

ClosureSurface	
Name:	ClosureSurface
Subtype of:	BoundarySurface
Definition:	ClosureSurfaces are used for buildings which are not enclosed completely, for example airplane hangars or barns. In order to represent those objects as geometrically closed volume object, the open sides are sealed (virtually closed) by ClosureSurfaces.
Description:	NOTE: Those special surfaces are taken into account, when needed to compute volumes and are neglected, when they are irrelevant or not appropriate, for example in visualisations.
Stereotypes:	«featureType»

5.8.2.1.6. Door

Door	
Name:	Door
Subtype of:	Opening
Definition:	The doors in the exterior shell of a building or between adjacent rooms. Doors can be used by people to enter or leave a building or room.
Description:	NOTE: When using LoD2 or LoD3 of CityGML (as indicated in INSPIRE Extended3D profile), the feature type Door is limited to the doors in the exterior shell of the building. Source: adapted from City GML.
Stereotypes:	«featureType»

5.8.2.1.7. GroundSurface

GroundSurface	
Name:	GroundSurface
Subtype of:	BoundarySurface
Definition:	A spatial object representing the ground plate of a building or building part. The polygon defining the ground plate is congruent with the building footprint.
Description:	Source: adapted from City GML

GroundSurface	
Stereotypes:	«featureType»

5.8.2.1.8. *Installation*

Installation	
Name:	Installation
Subtype of:	AbstractInstallation
Definition:	An external construction (of small size) or an external device serving the building or building part.
Description:	EXAMPLES: stairway, solar panel, external lift
Stereotypes:	«featureType»
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	GM_Primitive
Definition:	2D or 2.5D geometric representation.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: geometry3D	
Name:	Geometry 3D
Value type:	GM_Object
Definition:	3D geometric representation.
Multiplicity:	1
Association role: appearance	
Value type:	ParameterizedTexture
Definition:	The texture(s) attached to the installation and giving it its appearance.
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: boundedBy	
Value type:	BoundarySurface
Definition:	The surface(s) bounding the installation.
Multiplicity:	0..*
Stereotypes:	«voidable»

5.8.2.1.9. *InteriorInstallation*

InteriorInstallation	
Name:	InteriorInstallation
Definition:	An internal construction (generally of small size) or an internal device, serving the building or building part
Description:	EXAMPLE: stairs, lift
Stereotypes:	«featureType»
Attribute: inspireId	
Name:	inspireId
Value type:	Identifier
Definition:	External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.
Description:	NOTE1 External object identifiers are distinct from thematic object identifiers. NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object. NOTE 3 The unique identifier will not change during the life-time of a spatial

InteriorInstallation	
Multiplicity:	object. 0..1
Stereotypes:	«voidable»
Attribute: geometry3D	
Name:	geometry3D
Value type:	GM_Object
Definition:	3D geometric representation of the internal installation.
Multiplicity:	1
Attribute: installationNature	
Name:	installationNature
Value type:	InternalInstallationNatureValue
Definition:	The nature of the internal installation.
Multiplicity:	1
Attribute: beginLifespanVersion	
Name:	beginLifespanVersion
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
Multiplicity:	1
Stereotypes:	«voidable,lifeCycleInfo»
Attribute: endLifespanVersion	
Name:	endLifespanVersion
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
Multiplicity:	0..1
Stereotypes:	«voidable,lifeCycleInfo»
5.8.2.1.10. <i>Opening</i>	
Opening	
Name:	Opening
Definition:	The feature type <i>Opening</i> is the (abstract) base class for semantically describing openings like doors or windows in outer or inner walls. Openings only exist in models of LoD3 or LoD4.
Description:	NOTE: when using LoD3 of CityGML (as indicated in this INSPIRE Extended3D application schema), the spatial object type Opening is limited to the openings in the outer walls of the building Source : adapted from CityGML
Stereotypes:	«featureType»
Attribute: inspireId	
Name:	inspireId
Value type:	Identifier
Definition:	External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.
Description:	NOTE1 External object identifiers are distinct from thematic object identifiers. NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object. NOTE 3 The unique identifier will not change during the life-time of a spatial object.

Opening	
Multiplicity:	1
Attribute: geometry3D	
Name:	geometry3D
Value type:	BoundaryGeometry3D
Definition:	3D geometric representation of the opening (surface embedded in 3D).
Multiplicity:	1
Attribute: beginLifespanVersion	
Name:	beginLifespanVersion
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
Multiplicity:	1
Stereotypes:	«voidable,lifeCycleInfo»
Attribute: endLifespanVersion	
Name:	endLifespanVersion
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
Multiplicity:	0..1
Stereotypes:	«voidable,lifeCycleInfo»

5.8.2.1.11. OtherConstruction

OtherConstruction	
Name:	Other construction
Subtype of:	AbstractOtherConstruction
Definition:	An OtherConstruction is a self-standing construction that belongs to theme Buildings and that is not a Building.
Description:	NOTE 1: the main difference between a building and an other construction is the fact that an other construction does not need to be enclosed. NOTE 2: the other constructions to be considered under scope of theme buildings are the constructions that are not present in another INSPIRE theme and that are necessary for environmental use cases, such as the ones considered in this data specification. EXAMPLES: bridge, acoustic fence, city wall.
Stereotypes:	«featureType»
Attribute: geometry2D	
Name:	Geometry 2D
Value type:	GM_Primitive
Definition:	2D or 2.5D geometric representation.
Multiplicity:	0..*
Stereotypes:	«voidable»
Attribute: geometry3D	
Name:	Geometry 3D
Value type:	GM_Object
Definition:	3D geometric representation.
Multiplicity:	1
Association role: appearance	
Value type:	ParameterizedTexture
Definition:	The texture(s) attached to the other construction and giving it its appearance.

OtherConstruction	
Multiplicity:	0..*
Stereotypes:	«voidable»
Association role: boundedBy	
Value type:	BoundarySurface
Definition:	The surface(s) bounding the other construction.
Multiplicity:	0..*
Stereotypes:	«voidable»

5.8.2.1.12. OuterCeilingSurface

OuterCeilingSurface	
Name:	OuterCeilingSurface
Subtype of:	BoundarySurface
Definition:	A surface (feature) belonging to the outer building shell and having the orientation pointing downwards.
Description:	EXAMPLES: Visible part of the ceiling of a loggia or of the ceiling of a passage.
Stereotypes:	«featureType»

5.8.2.1.13. OuterFloorSurface

OuterFloorSurface	
Name:	OuterFloorSurface
Subtype of:	BoundarySurface
Definition:	A surface (feature) belonging to the outer building shell and with the orientation pointing upwards.
Description:	EXAMPLES: Visible part of the floor of a passage or of the floor of a loggia.
Stereotypes:	«featureType»

5.8.2.1.14. ParameterizedTexture

ParameterizedTexture	
Name:	ParameterizedTexture
Definition:	Texture representing the appearance aspect of a surface in the exterior shell of the building. The feature type <i>ParameterizedTexture</i> describes texture that is mapped to a surface (target).
Description:	SOURCE: adapted from CityGML.
Stereotypes:	«featureType»

Attribute: imageURI

Name:	imageURI
Value type:	URI
Definition:	Uniform Resource Identifier; gives indication where the image used for the texture may be found.
Multiplicity:	1

Attribute: mimeType

Name:	mimeType
Value type:	MimeTypeValue
Definition:	Format of the image used for texture.
Multiplicity:	1

Attribute: textureType

Name:	textureType
Value type:	TextureTypeValue
Definition:	Type of the texture; gives indication if the texture comes from real-world images or from standards images in libraries.

ParameterizedTexture	
Multiplicity:	1

5.8.2.1.15. RoofSurface

RoofSurface	
Name:	RoofSurface
Subtype of:	BoundarySurface
Definition:	The surfaces delimiting major roof parts of a building or building part are expressed by the feature type <i>RoofSurface</i> .
Description:	Source: adapted from CityGML
Stereotypes:	«featureType»
Attribute: materialOfRoof	
Name:	materialOfRoof
Value type:	MaterialOfRoofValue
Definition:	Material(s) of the building roof.
Multiplicity:	0..*
Stereotypes:	«voidable»

5.8.2.1.16. Room

Room	
Name:	Room
Definition:	A room is any distinguishable space within a structure. Usually, a room is separated from other spaces by interior walls; moreover, it is separated from outdoor areas by an exterior wall, sometimes with a door.
Stereotypes:	«featureType»
Attribute: inspireId	
Name:	inspireId
Value type:	Identifier
Definition:	External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.
Description:	NOTE1 External object identifiers are distinct from thematic object identifiers. NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object. NOTE 3 The unique identifier will not change during the life-time of a spatial object.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: geometry3D	
Name:	geometry3D
Value type:	GM_Solid
Definition:	3D geometric representation of the room.
Multiplicity:	1
Attribute: roomNature	
Name:	roomNature
Value type:	RoomNatureValue
Definition:	The nature (intended use or function) of the room.
Multiplicity:	0..1
Attribute: beginLifespanVersion	
Name:	beginLifespanVersion
Value type:	DateTime

Room	
Definition:	Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
Multiplicity:	1
Stereotypes:	«voidable,lifeCycleInfo»
Attribute: endLifespanVersion	
Name:	endLifespanVersion
Value type:	DateTime
Definition:	Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
Multiplicity:	0..1
Stereotypes:	«voidable,lifeCycleInfo»

5.8.2.1.17. WallSurface

WallSurface	
Name:	WallSurface
Subtype of:	BoundarySurface
Definition:	The surfaces that are parts of the building facade visible from the outside.
Description:	Source: adapted from CityGML.
Stereotypes:	«featureType»
Attribute: materialOfWall	
Name:	materialOfWall
Value type:	MaterialOfFacadeValue
Definition:	Material(s) of the building exterior walls.
Multiplicity:	0..1
Stereotypes:	«voidable»

5.8.2.1.18. Window

Window	
Name:	Window
Subtype of:	Opening
Definition:	Windows in the exterior shell of a building, or hatches between adjacent rooms.
Description:	NOTE 1: The formal difference between the classes window and door is that in normal cases windows are not specifically intended for the transit of people or vehicles. NOTE 2: when using LoD3 of CityGML (as indicated in INSPIRE Extended3D application schema), the feature type Window is limited to the windows in the exterior shell of the building. Source : adapted from CityGML.
Stereotypes:	«featureType»

5.8.2.2. Data types

5.8.2.2.1. BoundaryGeometry3D

BoundaryGeometry3D	
Name:	BoundaryGeometry3D
Definition:	The information related to the boundary geometry as 3D data.
Stereotypes:	«dataType»
Attribute: horizontalGeometryEstimatedAccuracyLoD2	
Name:	horizontalGeometryEstimatedAccuracyLoD2
Value type:	Length
Definition:	The estimated absolute positional accuracy of the (X,Y) coordinates of the LoD2 boundary representation, in the INSPIRE official Coordinate Reference System.

BoundaryGeometry3D	
Description:	<p>Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.</p> <p>NOTE: this mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.</p>
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: horizontalGeometryEstimatedAccuracyLoD3	
Name:	horizontalGeometryEstimatedAccuracyLoD3
Value type:	Length
Definition:	The estimated absolute positional accuracy of the (X,Y) coordinates of the LoD3 boundary representation, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE: this mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: horizontalGeometryEstimatedAccuracyLoD4	
Name:	horizontalGeometryEstimatedAccuracyLoD4
Value type:	Length
Definition:	The estimated absolute positional accuracy of the (X,Y) coordinates of the LoD4 boundary representation, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE: this mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: LoD2MultiSurface	
Name:	LoD2MultiSurface
Value type:	GM_MultiSurface
Definition:	The geometry of the boundary, corresponding to the LoD2 of CityGML.
Description:	LoD2 of City GML implies generalised geometry with vertical walls and simple roof shapes.
Multiplicity:	0..1
Attribute: LoD3MultiSurface	
Name:	LoD3MultiSurface
Value type:	GM_MultiSurface
Definition:	The geometry of the boundary, corresponding to the LoD3 of CityGML.
Description:	LoD3 of City GML represents the exact geometry of the building, approximating its true shape.
Multiplicity:	0..1

BoundaryGeometry3D	
Attribute: LoD4MultiSurface	
Name:	LoD4MultiSurface
Value type:	GM_MultiSurface
Definition:	The geometry of the boundary, corresponding to the LoD4 of CityGML.
Description:	LoD4 of City GML represents the accurate geometry of the building, approximating its true shape.
Multiplicity:	0..1
Attribute: verticalGeometryEstimatedAccuracyLoD2	
Name:	verticalGeometryEstimatedAccuracyLoD2
Value type:	Length
Definition:	The estimated absolute positional accuracy of the Z coordinate of the LoD2 boundary representation, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position. -- NOTE:This mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: verticalGeometryEstimatedAccuracyLoD3	
Name:	verticalGeometryEstimatedAccuracyLoD3
Value type:	Length
Definition:	The estimated absolute positional accuracy of the Z coordinate of the LoD3 boundary representation, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE:This mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Attribute: verticalGeometryEstimatedAccuracyLoD4	
Name:	verticalGeometryEstimatedAccuracyLoD4
Value type:	Length
Definition:	The estimated absolute positional accuracy of the Z coordinate of the LoD4 boundary representation, in the INSPIRE official Coordinate Reference System. Absolute positional accuracy is defined as the mean value of the positional uncertainties for a set of positions where the positional uncertainties are defined as the distance between a measured position and what is considered as the corresponding true position.
Description:	NOTE:This mean value may come from quality measures on a homogeneous population of buildings or from an estimation based on the knowledge of the production processes and of their accuracy.
Multiplicity:	0..1
Stereotypes:	«voidable»
Constraint: atLeastOneMandatoryGeometry	
Natural language:	

BoundaryGeometry3D

OCL: At least one of the mandatory geometries must be provided:

5.8.2.2.2. *TextCoordGen*

TextCoordGen

Name: TextCoordGen
Subtype of: TextureParametrization
Definition: Way to map a texture file (2D image coordinates) to a surface of the exterior shell of a building (3D real-world coordinates), by applying a transformation between the two coordinate reference systems.
Stereotypes: «dataType»

Attribute: worldToTexture

Name: worldToTexture
Value type: TransformationMatrix3x4
Definition: Matrix of the transformation between the file coordinates in an image to the geographical coordinates.
Multiplicity: 1

5.8.2.2.3. *TextCoordList*

TextCoordList

Name: TextCoordList
Subtype of: TextureParametrization
Definition: Way to map a texture file to a surface of the exterior shell of a building, by explicitly relating the coordinates of the image to the corresponding coordinates on the surface in the building shell.
Stereotypes: «dataType»

Attribute: ring

Name: ring
Value type: URI
Definition: Uniform Resource Identifier; gives indication where the ring (limit of image) may be found.
Multiplicity: 1

Attribute: textureCoordinates

Name: textureCoordinates
Value type: Number
Definition: List of coordinates in the texture file.
Multiplicity: 0..*

5.8.2.2.4. *TextureParametrization*

TextureParametrization (abstract)

Name: TextureParametrization
Definition: TextureParametrization is the abstract supertype of TextCoordGen and TextCoordList. It is used to relate both to a ParametrizedTexture.
Stereotypes: «dataType»

Association role: texture

Value type: ParameterizedTexture
Definition: The texture to be applied to the spatial object.
Multiplicity: 0..*

5.8.2.2.5. *TransformationMatrix3x4*

TransformationMatrix3x4

TransformationMatrix3x4	
Name:	TransformationMatrix3x4
Definition:	A matrix providing the transformation function between coordinates of the texture image and the geographical coordinates.
Stereotypes:	«dataType»
Attribute: elements	
Name:	elements
Value type:	Number
Definition:	The matrix elements.
Multiplicity:	0..*

5.8.2.3. Code lists

5.8.2.3.1. *InternalInstallationNatureValue*

InternalInstallationNatureValue	
Name:	InternalInstallationNatureValue
Definition:	Code list for the possible values of the nature of an internal installation.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/InternalInstallationNatureValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.8.2.3.2. *MimeTypeValue*

MimeTypeValue	
Name:	MimeTypeValue
Definition:	Mime types code list.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/MimeTypeValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.8.2.3.3. *RoomNatureValue*

RoomNatureValue	
Name:	RoomNatureValue
Definition:	Code list giving the possible values for the nature of a room (use or intended function).
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/RoomNatureValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.8.2.3.4. *TextureTypeValue*

TextureTypeValue	
Name:	TextureTypeValue
Definition:	The texture type code list.
Extensibility:	open
Identifier:	http://inspire.ec.europa.eu/codelist/TextureTypeValue
Values:	The allowed values for this code list comprise the values specified in <i>Annex C</i> and additional values at any level defined by data providers. <i>Annex C</i> includes recommended values that may be used by data providers.

5.8.2.4. Imported types (informative)

This section lists definitions for feature types, data types and enumerations and code lists that are defined in other application schemas. The section is purely informative and should help the reader understand the feature catalogue presented in the previous sections. For the normative documentation of these types, see the given references.

5.8.2.4.1. *AbstractBuildingUnit*

AbstractBuildingUnit (abstract)	
Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the semantic properties of building units. A BuildingUnit is a subdivision of Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc.
Description:	Building units are spatial objects aimed at subdividing buildings and/or building parts into smaller parts that are treated as separate entities in daily life. A building unit is homogeneous, regarding management aspects. EXAMPLES: It may be e.g. an apartment in a condominium, a terraced house, or a shop inside a shopping arcade. NOTE 1: According to national regulations, a building unit may be a flat, a cellar, a garage or set of a flat, a cellar and a garage. NOTE 2: According to national regulation, a building that is one entity for daily life (typically, a single family house) may be considered as a Building composed of one BuildingUnit or as a Building composed of zero BuildingUnit.

5.8.2.4.2. *AbstractInstallation*

AbstractInstallation (abstract)	
Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the semantic properties of installations. An external construction (of small size) or an external device serving the building or building part.
Description:	EXAMPLES: stairway, solar panel, external lift

5.8.2.4.3. *AbstractOtherConstruction*

AbstractOtherConstruction (abstract)	
Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the semantic properties of other constructions. An other construction is a self-standing construction that belongs to theme <i>Buildings</i> and that is not a Building.
Description:	NOTE 1: the main difference between a building and an other construction is the fact that an other construction does not need to be enclosed. NOTE 2: the other constructions to be considered under scope of theme <i>Buildings</i> are the constructions that are not present in another INSPIRE theme and that are necessary for environmental use cases, such as the ones considered in this data specification. EXAMPLES: bridge, acoustic fence, city wall.

5.8.2.4.4. *BuildingInfo*

BuildingInfo (abstract)	
Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Abstract spatial object type grouping the additional specific properties of Building and Building Part.
Description:	NOTE 1: The additional properties are those that are not already included in the

BuildingInfo (abstract)		
	base	application schema.
	NOTE 2: The specific properties are the properties that apply to Building and BuildingPart without applying to BuildingUnit.	

5.8.2.4.5. *DateTime*

DateTime		
Package:	Date and Time	
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]	

5.8.2.4.6. *GM_MultiSolid*

GM_MultiSolid		
Package:	Geometric aggregates	
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]	

5.8.2.4.7. *GM_MultiSurface*

GM_MultiSurface		
Package:	Geometric aggregates	
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]	

5.8.2.4.8. *GM_Object*

GM_Object (abstract)		
Package:	Geometry root	
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]	

5.8.2.4.9. *GM_Primitive*

GM_Primitive (abstract)		
Package:	Geometric primitive	
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]	

5.8.2.4.10. *GM_Solid*

GM_Solid		
Package:	Geometric primitive	
Reference:	Geographic information -- Spatial schema [ISO 19107:2003]	

5.8.2.4.11. *Identifier*

Identifier		
Package:	Base Types	
Reference:	INSPIRE Generic Conceptual Model, version 3.4 [DS-D2.5]	
Definition:	External unique object identifier published by the responsible body, which may be used by external applications to reference the spatial object.	
Description:	NOTE1 External object identifiers are distinct from thematic object identifiers.	
	NOTE 2 The voidable version identifier attribute is not part of the unique identifier of a spatial object and may be used to distinguish two versions of the same spatial object.	
	NOTE 3 The unique identifier will not change during the life-time of a spatial object.	

5.8.2.4.12. *Length*

Length		
Package:	Units of Measure	
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]	

5.8.2.4.13. *MaterialOfFacadeValue*

MaterialOfFacadeValue

Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Code list for the possible values of MaterialOfFacade

5.8.2.4.14. *MaterialOfRoofValue*

MaterialOfRoofValue

Package:	BuildingsExtendedBase
Reference:	INSPIRE Data specification on <i>Buildings</i> [DS-D2.8.III.2]
Definition:	Code list for possible values of attribute MaterialOfRoof

5.8.2.4.15. *Number*

Number (abstract)

Package:	Numerics
Reference:	Geographic information -- Conceptual schema language [ISO/TS 19103:2005]

5.8.2.4.16. *URI*

URI

Package:	basicTypes
Reference:	Geographic information -- Geography Markup Language (GML) [ISO 19136:2007]

