9 Delivery

9.1 Updates

IR Requirement
Article 8
Updates

1. Member States shall make available updates of data on a regular basis.
2. All updates shall be made available at the latest 6 months after the change was applied in the source data set, unless a different period is specified for a specific spatial data theme in Annex II.

NOTE In this data specification, no exception is specified, so all updates shall be made available at the latest 6 months after the change was applied in the source data set.

9.2 Delivery medium

According to Article 11(1) of the INSPIRE Directive, Member States shall establish and operate a network of services for INSPIRE spatial data sets and services. The relevant network service types for making spatial data available are:

- *view services* making it possible, as a minimum, to display, navigate, zoom in/out, pan, or overlay viewable spatial data sets and to display legend information and any relevant content of metadata;
- *download services*, enabling copies of spatial data sets, or parts of such sets, to be downloaded and, where practicable, accessed directly;
- *transformation services*, enabling spatial data sets to be transformed with a view to achieving interoperability.

NOTE For the relevant requirements and recommendations for network services, see the relevant Implementing Rules and Technical Guidelines[15].

EXAMPLE 1 Through the Get Spatial Objects function, a download service can either download a pre-defined data set or pre-defined part of a data set (non-direct access download service), or give direct access to the spatial objects contained in the data set, and download selections of spatial objects based upon a query (direct access download service). To execute such a request, some of the following information might be required:

– the list of spatial object types and/or predefined data sets that are offered by the download service (to be provided through the Get Download Service Metadata operation),
– and the query capabilities section advertising the types of predicates that may be used to form a query expression (to be provided through the Get Download Service Metadata operation, where applicable),
– a description of spatial object types offered by a download service instance (to be provided through the Describe Spatial Object Types operation).

EXAMPLE 2 Through the Transform function, a transformation service carries out data content transformations from native data forms to the INSPIRE-compliant form and vice versa. If this operation is directly called by an application to transform source data (e.g. obtained through a download service) that is not yet conformant with this data specification, the following parameters are required:
Input data (mandatory). The data set to be transformed.

[15]The Implementing Rules and Technical Guidelines on INSPIRE Network Services are available at http://inspire.jrc.ec.europa.eu/index.cfm/pageid/5

- Source model (mandatory, if cannot be determined from the input data). The model in which the input data is provided.
- Target model (mandatory). The model in which the results are expected.
- Model mapping (mandatory, unless a default exists). Detailed description of how the transformation is to be carried out.

9.3 Encodings

The IRs contain the following two requirements for the encoding to be used to make data available.

IR Requirement
Article 7
Encoding

1. Every encoding rule used to encode spatial data shall conform to EN ISO 19118. In particular, it shall specify schema conversion rules for all spatial object types and all attributes and association roles and the output data structure used.
2. Every encoding rule used to encode spatial data shall be made available.

NOTE ISO 19118:2011 specifies the requirements for defining encoding rules used for interchange of geographic data within the set of International Standards known as the “ISO 19100 series”. An encoding rule allows geographic information defined by application schemas and standardized schemas to be coded into a system-independent data structure suitable for transport and storage. The encoding rule specifies the types of data being coded and the syntax, structure and coding schemes used in the resulting data structure. Specifically, ISO 19118:2011 includes
- requirements for creating encoding rules based on UML schemas,
- requirements for creating encoding services, and
- requirements for XML-based encoding rules for neutral interchange of data.

While the IRs do not oblige the usage of a specific encoding, these Technical Guidelines propose to make data related to the spatial data theme *Buildings* available at least in the default encoding(s) specified in section 9.3.1. In this section, a number of TG requirements are listed that need to be met in order to be conformant with the default encoding(s).

The proposed default encoding(s) meet the requirements in Article 7 of the IRs, i.e. they are conformant with ISO 19118 and (since they are included in this specification) publicly available.

9.3.1 Default Encoding(s)

9.3.1.1. Specific requirements for GML encoding

This data specification proposes the use of GML as the default encoding, as recommended in sections 7.2 and 7.3 of [DS-D2.7]. GML is an XML encoding in compliance with ISO 19118, as required in Article 7(1). For details, see [ISO 19136], and in particular Annex E (UML-to-GML application schema encoding rules).

The following TG requirements need to be met in order to be conformant with GML encodings.

TG Requirement 6 Data instance (XML) documents shall validate without error against the provided XML schema.

NOTE 1 Not all constraints defined in the application schemas can be mapped to XML. Therefore, the following requirement is necessary.

NOTE 2 The obligation to use only the allowed code list values specified for attributes and most of the constraints defined in the application schemas cannot be mapped to the XML sch. They can therefore not be enforced through schema validation. It may be possible to express some of these constraints using other schema or rule languages (e.g. Schematron), in order to enable automatic validation.

9.3.1.2. Default encoding(s) for application schema BuildingsBase

Name: BuildingsBase GML Application Schema
Version: version 3.0
Specification: D2.8.III.2 Data Specification on *Buildings* – Technical Guidelines
Character set: UTF-8

The xml schema document is available on the INSPIRE website
http://inspire.ec.europa.eu/schemas/bu-base/3.0/BuildingsBase.xsd.

9.3.1.3. Default encoding(s) for application schema Buildings2D

Name: Buildings2D GML Application Schema
Version: version 3.0
Specification: D2.8.III.2 Data Specification on *Buildings* – Technical Guidelines
Character set: UTF-8

The xml schema document is available on the INSPIRE website
http://inspire.ec.europa.eu/schemas/bu-core2d/3.0/BuildingsCore2D.xsd.

9.3.1.4. Default encoding(s) for application schema Buildings3D

Name: Buildings3D GML Application Schema
Version: version 3.0
Specification: D2.8.III.2 Data Specification on *Buildings* – Technical Guidelines
Character set: UTF-8

The xml schema document is available on the INSPIRE website
http://inspire.ec.europa.eu/schemas/bu-core3d/3.0/BuildingsCore3D.xsd.

9.3.1.5. Default encoding(s) for application schema BuildingsExtendedBase

Name: BuildingsExtendedBase GML Application Schema
Version: version 3.0
Specification: D2.8.III.2 Data Specification on *Buildings* – Technical Guidelines
Character set: UTF-8

The xml schema document is available on the INSPIRE website
http://inspire.ec.europa.eu/draft-schemas/bu-ext/3.0/BuildingsExtendedBase.xsd.

9.3.1.6. Default encoding(s) for application schema BuildingsExtended2D

Name: BuildingsExtended2D GML Application Schema
Version: version 3.0
Specification: D2.8.III.2 Data Specification on *Buildings* – Technical Guidelines
Character set: UTF-8

The xml schema document is available on the INSPIRE website
http://inspire.ec.europa.eu/schemas/draft-schemas/bu-ext2d/3.0/BuildingsExtended2D.xsd.

9.3.1.7. Default encoding(s) for application schema BuildingsExtended3D

Name: BuildingsExtended3D GML Application Schema
Version: version 3.0
Specification: D2.8.III.2 Data Specification on *Buildings* – Technical Guidelines
Character set: UTF-8

The xml schema document is available on the INSPIRE website
http://inspire.ec.europa.eu/schemas/draft-schemas/bu-ext3d/3.0/BuildingsExtended3D.xsd.

9.3.2 Recommended Encoding(s)

Recommendation 34	It is recommended that also the encodings specified in this section be provided for the relevant application schemas.

9.3.2.1. Alternative encoding for application schema BuildingsExtended3D

9.3.2.1.1. Context- rationale
The two 3D profiles of the INSPIRE building model (Core 3D profile and Extended 3D profile) share many properties with CityGML, particularly the representation of the 3D geometry. Currently, there are many tools which support CityGML (see Gröger & Plümer 2012 or *www.citygmlwiki.org* for details): data capturing tools providing output data in CityGML format, visualization tools which offer CityGML as input format, tools for checking consistency, Web Feature Services which read and provide CityGML, and data bases which store data based on CityGML and provide corresponding interfaces. Due to an increasing international usage and availability of CityGML in many cities and municipalities in Europe, the Middle East and Asia as well as in various organizations and universities all over Europe this list of tools is expected to grow significantly in the future.

However, these CityGML tools cannot be used directly for INSPIRE building data, since the standard encodings of the four INSPIRE profiles differ from the CityGML encoding, even if the conceptual INSPIRE application schemas widely use the CityGML concepts. In order to utilize these tools also for INSPIRE building data, an alternative encoding based on CityGML is provided here.

CityGML provides a mechanism for adding application specific extensions to the CityGML model. This mechanism is called Application Domain Extension (ADE). An ADE is a sub-schema of CityGML through which new attributes, geometries, and relations can be added to existing CityGML classes such as the number of dwellings to the class AbstractBuilding. Moreover, also new classes can be added to the CityGML model such as OtherConstruction. These new classes typically are sub-classes of appropriate CityGML classes (e.g., of Building or of the topmost class CityObject). An ADE is an application schema in its own namespace, which imports the XML schema definitions of relevant CityGML modules. Formally, an ADE is specified as an XML schema, enabling the validation of instance documents against this schema. Examples for existing CityGML ADEs are the NoiseADE for noise simulation and mapping, the facility management ADE, the UtilityNetworkADE, and the HydroADE for hydrographical applications. Furthermore, the new national geospatial standard in the Netherlands which integrates 2D and 3D data is implemented as CityGML ADE.

9.3.2.1.2. Narrative description
A CityGML ADE currently is provided for the Core 3D profile (application schema Buildings3D) of the INSPIRE building model. An ADE for the Extended 3D profile (application schema BuildingsExtended3D) is in preparation. In these ADEs, those properties which are shared by INSPIRE and CityGML are inherited from CityGML, while properties which are different or additional in INSPIRE are defined in the CityGML ADE for INSPIRE. Constraints are employed to describe these definitions and settings. For more details on the CityGML ADE for the INSPIRE Core 3D profile, see Gröger, Kutzner & Kolbe (2013).

The UML diagram of the CityGML ADE for the Core 3D profile is depicted in Figure 60 and Figure 61, and the conceptual mapping between the UML classes Building and BuildingPart of the INSPIRE Core

3D profile and their encoding (GML object elements and types) in Table 10. In the left part of Figure 60, the classes AbstractBuilding, Building and BuildingPart from the CityGML Building module are shown. The attributes and relations which are specific to INSPIRE are provided by the ADE class AbstractBuilding in the INSPIRE::ADE::Core3D package. These attributes and relations are added to the CityGML class AbstractBuilding (and hence, to its sub-classes Building and BuildingPart) in the CityGML::Building package by the so-called hooking mechanism. This mechanism is indicated by the generalization relation marked with the stereotype «ADE»: the triangle points to the class receiving the attributes and relations from the ADE class AbstractBuilding at the other side of the generalization relation which is labeled with the stereotype «ADEElement». Hence, all attributes and relations of the ADE class AbstractBuilding (in ADE namespace) are added to the CityGML class AbstractBuilding (CityGML namespace). The approach how to represent the hooking mechanism in UML is described in van den Brink, Stoter & Zlatanova (2012).

Those attributes and relations which are common to both the CityGML building module and the INSPIRE Building module (with regard to semantics and data types) are represented by CityGML. An example is the number of storeys above ground (storeysAboveGround). Those attributes and relations which are specific to INSPIRE and for which no CityGML counterpart is available, or which differ from CityGML with regard to semantics or data type, are represented by the CityGML ADE for Core 3D (class AbstractBuilding). The constraint 'Mapping of INSPIRE *Buildings* to CityGML or to the INSPIRE Core 3D ADE' included in the UML diagram as well as Table 11 define this mapping. The INSPIRE properties and data types which represent geometry (properties geometry3DLoD1, geometry3DLoD2, …, data types BuildingGeometry3DLoD1, …) have been split: The pure 3D geometry is represented by CityGML properties (lod1Solid, lod2Solid, …), whereas metadata such as accuracy is provided by the INSPIRE data types mgeometry3DMetadataLoD1, geometry3DMetadataLoD1, …), c.f. Figure 61. In CityGML, a BuildingPart may again have BuildingParts, whereas in INSPIRE parts of parts are not allowed. Furthermore, in CityGML there is no mandatory geometry in the class AbstractBuilding. This is in contrast to INSPIRE, where BuildingParts and *Buildings* without BuildingParts must have at least one geometry. Therefore, two constrains have been added to enforce the INSPIRE behavior.

Note that in the UML diagram, the classes Building and BuildingPart are not necessary for deriving the ADE; they have been included only to make the diagram more understandable.
As an example for the CityGML ADE encoding for the Core 3D profile, an instance document representing a simple building is given in Figure 63.

The ADE provides a practicable alternative to the native INSPIRE encoding, enabling INSPIRE building data not only to be identified and processed by currently available CityGML software as CityGML Building objects, but also to be combined with other ADEs extending the CityGML Building feature type. Furthermore, the conceptual mapping between the INSPIRE Core 3D profile and the CityGML ADE for Core 3D as described in Table 10 and Table 11 can serve in deriving INSPIRE building data directly from CityGML data and vice versa, creating views in this way. This means, when CityGML data is imported into an INSPIRE data base, the conceptual mapping describes which data have to be read and processed by the importing tool and which data can be skipped. The same holds true when INSPIRE building data is exported to CityGML data.

The INSPIRE Core 3D ADE has been developed by the TWG on *Buildings* in cooperation with the Chair of Geoinformatics at Technische Universität München and the Chair of Geoinformation, Institute for Geodesy and Geoinformation at University of Bonn, Germany.

The CityGML ADE for the INSPIRE Extended 3D profile is under preparation and will be supplied and tested by the wider stakeholder community as part of the INSPIRE Maintenance and Implementation Framework.

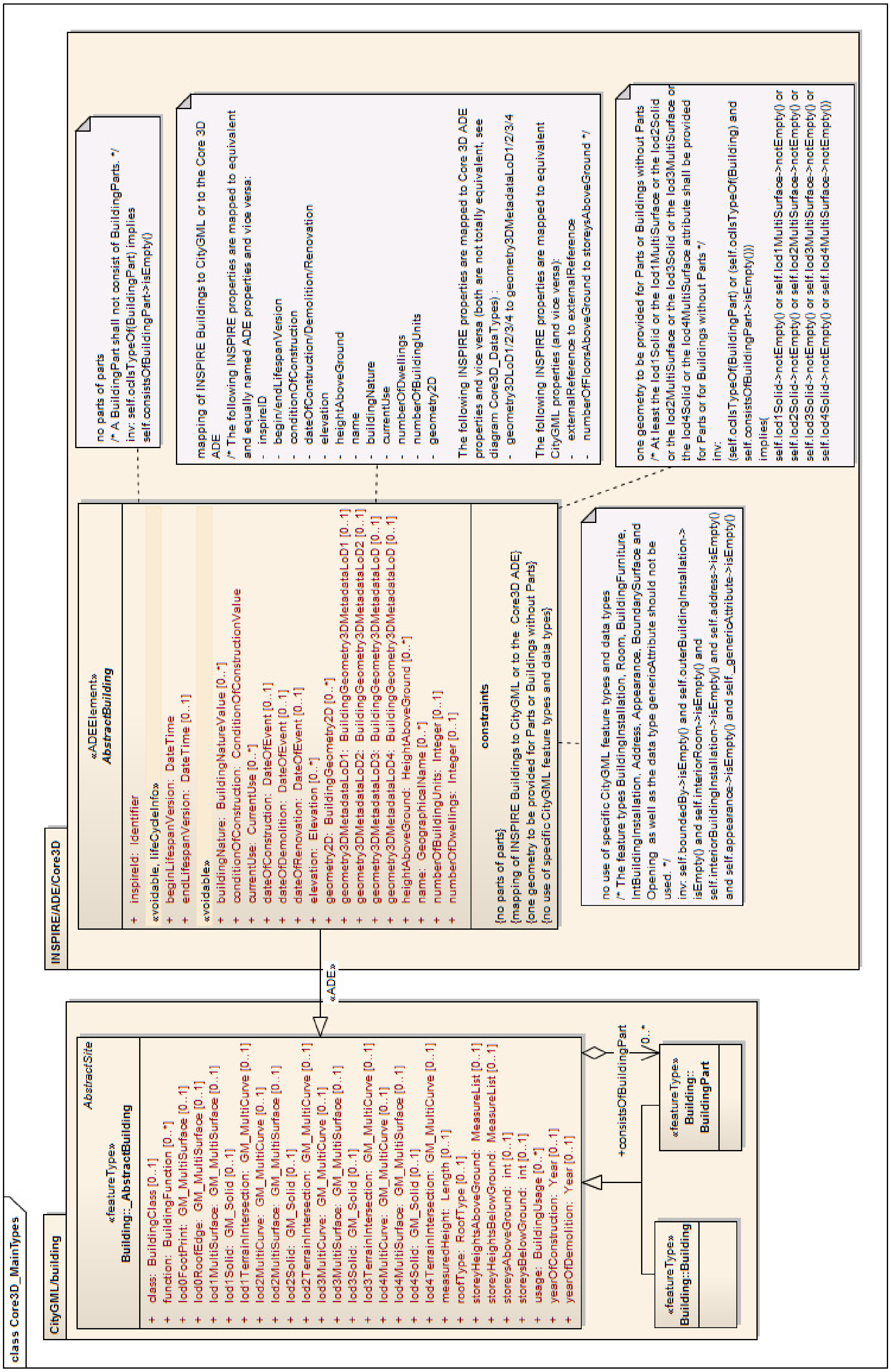


Figure 60: UML diagram (main types) of the CityGML ADE for the Core 3D profile as an alternative encoding.

Table 10: Mappings between conceptual UML classes and the associated GML object elements, ML Schema types and GML property types

UML class	GML element	GML type	GML property
INSPIRE:: Core3D::Building	CityGML/building: Building	CityGML/building: BuildingType	n. a.
INSPIRE:: Core3D::BuildingPart	CityGML/building: BuildingPart	CityGML/building: BuildingPartType	n. a.

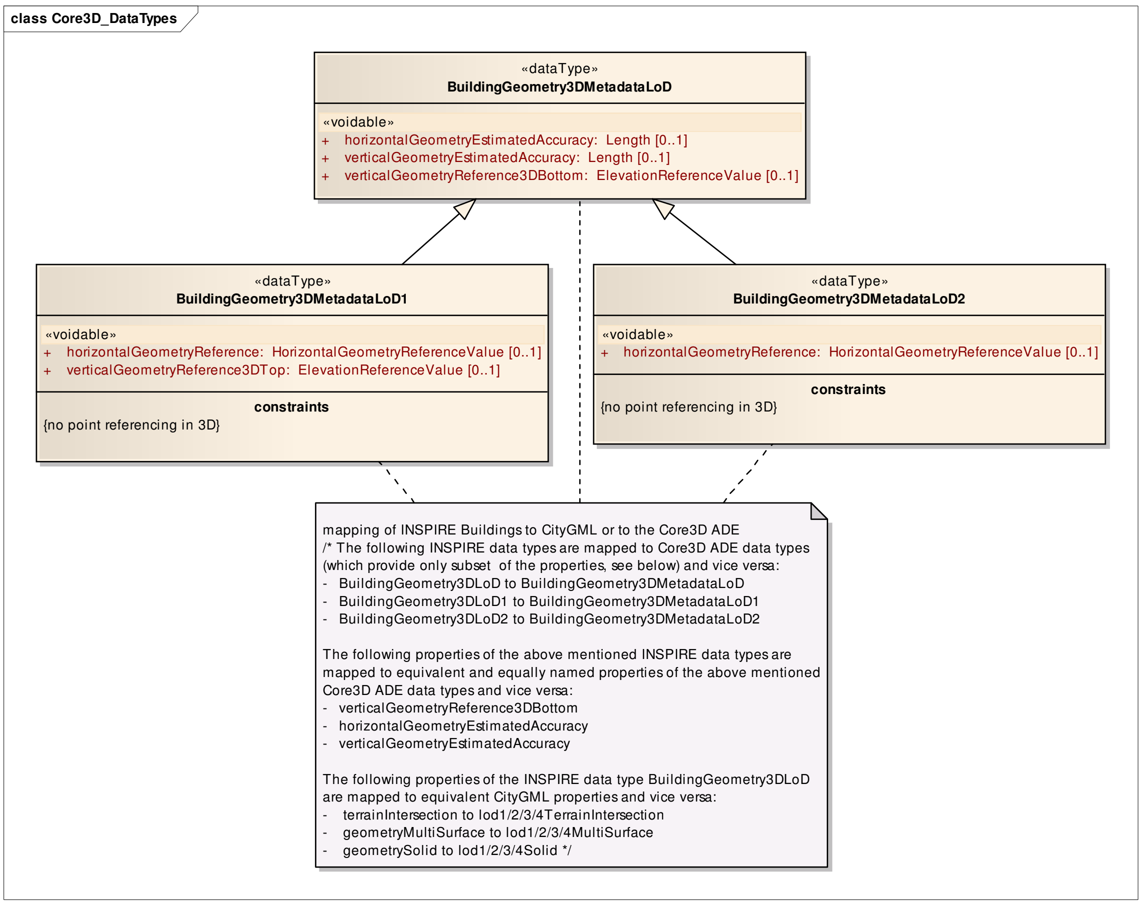


Figure 61: UML diagram (data types) of the CityGML ADE for INSPIRE Core 3D profile as an alternative encoding.

Table 11: Conceptual mapping between the UML classes, attributes and relations of the INSPIRE Core3D profile and the associated UML classes, attributes and relations of the INSPIRE Core3D CityGML ADE

UML class in the INSPIRE Core 3D profile	UML attribute/ relation (in INSPIRE application schema)	UML class in CityGML / CityGML ADE for INSPIRE Core 3D profile	UML attribute/relation in CityGML / CityGML ADE for INSPIRE Core 3D profile
BuildingsBase:: AbstractConstruction	inspireId	INSPIRE::ADE::Core3D:: AbstractBuilding	inspireId
BuildingsBase:: AbstractConstruction	beginLifespan Version	INSPIRE::ADE::Core3D:: AbstractBuilding	beginLifespanVersion

BuildingsBase:: AbstractConstruction	endLifespan Version	INSPIRE::ADE::Core3D:: AbstractBuilding	endLifespanVersion
BuildingsBase:: AbstractConstruction	conditionOf Construction	INSPIRE::ADE::Core3D:: AbstractBuilding	conditionOfConstruction
BuildingsBase:: AbstractConstruction	dateOf Construction	INSPIRE::ADE::Core3D:: AbstractBuilding	dateOfConstruction
BuildingsBase:: AbstractConstruction	dateOf Demolition	INSPIRE::ADE::Core3D:: AbstractBuilding	dateOfDemolition
BuildingsBase:: AbstractConstruction	dateOf Renovation	INSPIRE::ADE::Core3D:: AbstractBuilding	dateOfRenovation
BuildingsBase:: AbstractConstruction	elevation	INSPIRE::ADE::Core3D:: AbstractBuilding	Elevation
BuildingsBase:: AbstractConstruction	external Reference	CityGML::Core:: AbstractCityObject	externalReference
BuildingsBase:: AbstractConstruction	heightAbove Ground	INSPIRE::ADE::Core3D:: AbstractBuilding	heightAboveGround
BuildingsBase:: AbstractConstruction	Name	INSPIRE::ADE::Core3D:: AbstractBuilding	Name
BuildingsBase:: AbstractBuildung	buildingNature	INSPIRE::ADE::Core3D:: AbstractBuilding	buildingNature
BuildingsBase:: AbstractBuildung	currentUse	INSPIRE::ADE::Core3D:: AbstractBuilding	currentUse
BuildingsBase:: AbstractBuildung	numberOf Dwellings	INSPIRE::ADE::Core3D:: AbstractBuilding	numberOfDwellings
BuildingsBase:: AbstractBuildung	numberOf BuildingUnits	INSPIRE::ADE::Core3D:: AbstractBuilding	numberOfBuildingUnits
BuildingsBase:: AbstractBuildung	numberOf FloorsAbove Ground	CityGML::Building:: AbstractBuilding	storeysAboveGround
BuildingsBase:: Building	Parts	CityGML::Building:: AbstractBuilding	consistsOfBuildingPart
Buildingds3D:: Building	geometry3D LoD1	CityGML::Building:: AbstractBuilding	lod1Solid
		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD1
Buildingds3D:: Building	geometry3D LoD2	CityGML::Building:: AbstractBuilding	lod2Solid
		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD2
Buildingds3D:: Building	geometry3D LoD3	CityGML::Bbuilding:: AbstractBuilding	lod3Solid
		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD3
Buildingds3D:: Building	geometry3D LoD4	CityGML::Building:: AbstractBuildingType	lod4Solid
		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD4
Buildingds3D:: Building	geometry2D	INSPIRE::ADE::Core3D:: AbstractBuilding	geometry2D
Buildingds3D:: BuildingPart	geometry3D LoD1	CityGML::Building:: AbstractBuilding	lod1Solid
		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD1
Buildingds3D:: BuildingPart	geometry3D LoD2	CityGML/Building:: AbstractBuilding	lod2Solid
		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD2
Buildingds3D:: BuildingPart	geometry3D LoD3	CityGML::Building:: AbstractBuilding	lod3Solid

		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD3
Buildingds3D:: BuildingPart	geometry3D LoD4	CityGML::Building:: AbstractBuilding	lod4Solid
		INSPIRE::ADE::Core3D:: AbstractBuilding	geometry3DMetadata LoD4
Buildingds3D:: BuildingPart	geometry2D	INSPIRE::ADE::Core3D:: AbstractBuilding	geometry2D
Buildingds3D:: BuildingGeometry3D LoD	geometryMulti Surface	CityGML::building:: AbstractBuilding	lod1MultiSurface lod2MultiSurface lod3MultiSurface lod4MultiSurface
Buildingds3D:: BuildingGeometry3D LoD	geometrySolid	CityGML::Building:: AbstractBuilding	lod1Solid lod2Solid lod3Solid lod4Solid
Buildingds3D:: BuildingGeometry3D LoD	terrain Intersection:	CityGML::Building:: AbstractBuilding	lod1TerrainIntersection lod2TerrainIntersection lod3TerrainIntersection lod4TerrainIntersection
Buildingds3D:: BuildingGeometry3D LoD	Vertical Geometry Reference3D Bottom	INSPIRE::ADE::Core3D:: BuildingGeometry3D MetadataLoD	verticalGeometry Reference3DBottom
Buildingds3D:: BuildingGeometry3D LoD	horizontal Geometry Estimated Accuracy	INSPIRE::ADE::Core3D:: BuildingGeometry3D MetadataLoD	horizontalGeometry EstimatedAccuracy
Buildingds3D:: BuildingGeometry3D LoD	vertical Geometry Estimated Accuracy	INSPIRE::ADE::Core3D:: BuildingGeometry3D MetadataLoD	verticalGeometry EstimatedAccuracy
Buildingds3D:: BuildingGeometry3D LoD1	Horizontal Geometry Reference	INSPIRE::ADE::Core3D:: BuildingGeometry3D MetadataLoD1	horizontalGeometry Reference
Buildingds3D:: BuildingGeometry3D LoD1	Vertical Geometry Reference3D Top	INSPIRE::ADE::Core3D:: BuildingGeometry3D MetadataLoD1	verticalGeometry Reference3DTop
Buildingds3D:: BuildingGeometry3D LoD2	horizontal Geometry Reference	INSPIRE::ADE::Core3D:: BuildingGeometry3D MetadataLoD2	horizontalGeometry Reference

```xml
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel xsi:schemaLocation="http://www.citygml.org/ade/inspire/core3dcore3d.xsd
http://www.opengis.net/citygml/2.0 http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:bldg="http://www.opengis.net/citygml/building/2.0"
xmlns:gml="http://www.opengis.net/gml" xmlns:core="http://www.opengis.net/citygml/2.0"
xmlns:bu-3d="http://www.citygml.org/ade/inspire/core3d"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<gml:boundedBy>
    <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/7409">
      <gml:lowerCorner>370000.0 5805000.0</gml:lowerCorner>
      <gml:upperCorner>370249.0 5806000.0</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <core:cityObjectMember>
     <bldg:Building>
          <bldg:lod1Solid>
               <gml:Solid>
                 <gml:exterior>
                                 <gml:CompositeSurface>
                           <gml:surfaceMember>
                                <gml:Polygon gml:id="UUID_a988d2a9-d749-4380-abd6-bedae2061d05">
                                   <gml:exterior>
                                     <gml:LinearRing gml:id="UUID_8fb76b13-51cf-44ad-a13c-3984495211ee">
                                          <gml:posList srsDimension="3">
                                        370089.758 5805596.288 30.911
                                        370091.018 5805600.028 30.911
                                        370093.347 5805599.268 30.911
                                         </gml:posList>
                                        </gml:LinearRing>
                                  </gml:exterior>
                                </gml:Polygon>
               </gml:surfaceMember>
                   </gml:CompositeSurface>
                 </gml:exterior>
               </gml:Solid>
          </bldg:lod1Solid>
          <bu-3d:inspireId>
               <base:Identifier xmlns:base="urn:x-inspire:specification:gmlas:BaseTypes:3.2">
                 <base:localId> To be filled  </base:localId>
                 <base:namespace/>
               </base:Identifier>
          </bu-3d:inspireId>
          <bu-3d:conditionOfConstruction

xlink:href="http://inspire.ec.europa.eu/codelist/CondidtionOfConstructionValue/functional"/>
               <bu-3d:geometry3DMetadataLoD1>
                 <bu-3d:BuildingGeometry3DMetadataLoD1>
                   <bu-3d:verticalGeometryEstimatedAccuracy uom="m">0.05
                                </bu-3d:verticalGeometryEstimatedAccuracy>
                   <bu-3d:horizontalGeometryReference

xlink:href="http://inspire.ec.europa.eu/codelist/HorizontalGeometryReferenceValue/aboveGroundEnvelope"/>
               </bu-3d:BuildingGeometry3DMetadataLoD1>
          </bu-3d:geometry3DMetadataLoD1>
     </bldg:Building>
     </core:cityObjectMember>
</core:CityModel>
```

Figure 62:: Example instance document for the CityGML ADE for the INSPIRE Core 3D profile

9.4 Data Capture

This chapter does not aim to define data capture rules as INSPIRE is based on existing data. The data capture rules of source data are up to each data producer. Data producers should clearly describe any deviations from these guidelines in the metadata. This chapter aims to give some guidelines about how to match existing data to INSPIRE specifications. It is not exhaustive but focus on the aspects that are expected to raise some issues.

9.5 Scope of theme Buildings

9.5.1 Purpose

Existing data should be made compliant to INSPIRE, taking into account cost-benefit considerations. The scope of theme *Buildings* and definition of its core feature type Building are rather generic and may open the door to various interpretations.

The costs of transformation will depend on how data related to theme *Buildings* is organised within a Member State. For instance, some data producers have all constructions in a single feature type whereas other data producers have different feature types for buildings, for annex constructions, for urban furniture …. ; building related data may be scattered between various producers or may be under the responsibility of only one organisation. Making whole scope of theme *Buildings* compliant to INSPIRE will likely be easier when all data regarding buildings and constructions is within the same feature type or at least in the same data set.

This paragraph aims to clarify the interpretation of scope, to provide recommendations about which kinds of buildings and constructions are expected by INSPIRE and so, to assess the benefits of making data compliant to INSPIRE.

The general rules or priority are given by the modular scope defined in clause 2.2.

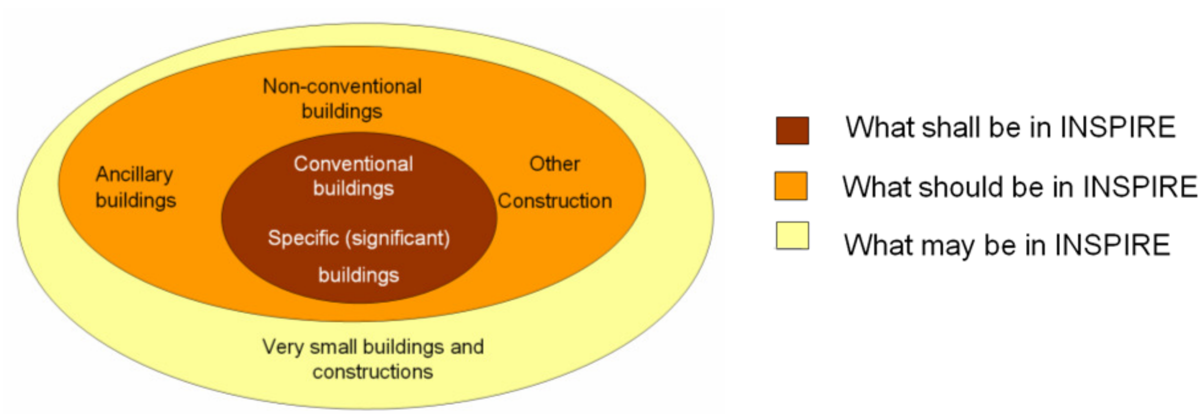


Figure 63: The modular approach for scope of theme Buildings

9.5.2 Code lists

The possible values provided in the code lists on current use and on nature of buildings and constructions, provide the general guidelines about what is expected by / should be in INSPIRE.

More accurately:

- The buildings hosting human activities, i.e. buildings whose current use is residential, agriculture, industrial, commerceAndservices are **necessary for** / **strongly expected** by many INSPIRE use cases.

- The buildings whose current use is ancillary are useful for / expected by some INSPIRE use cases.

- The buildings that may be obstacles or valuable landmarks for air traffic, i.e. those whose building nature is arch, dam, tower, lighthouse, silo, windmill, wind turbine, transformer, stadium, storageTank are **necessary for** / **strongly expected by** some INSPIRE use cases, air traffic being an international use case.

- The buildings whose building nature takes other values (e.g. shed, greenhouse, bunker, canopy, chapel, caveBuilding…) are useful for / expected by some INSPIRE use cases (landscape description, mapping).

- The other constructions are **also necessary for** / **strongly expected by** some INSPIRE use cases:
  - elevated constructions (crane, antenna, monument, pylons, …) as obstacles for air traffic
  - environmental barriers (protectiveStructure, acousticFence, retainingWall) or open air pools for mitigation of risk and of pollution
  - bridges and tunnels for planning of rescue itineraries in case of disaster.

NOTE: according to the modular scope, other constructions are under the second priority, due to the expected feasibility issues, but there are quite strong user requirements about these constructions.

Recommendation 7 OtherConstructions should be made available for INSPIRE, as much as possible.

9.5.3 Definition of theme buildings

Considered as under scope of the theme *Buildings* are **constructions** above and/or underground which are intended or used for the shelter of humans, animals, things, the production of economic goods or the delivery of services and that refer to any structure permanently constructed or erected on its site.

NOTE 1: According to the definition, the construction should be **permanently constructed or erected on its site.** However, the notion of "permanence" may be interpreted in a flexible way. For instance, some types of dwellings are theoretically designed to be mobile (e.g. mobile homes, caravans) or to be used for short time (huts, cabins, shacks, shanties) but are in practice used in permanent way and may require the set up of public services. Moreover, there are strong user requirements for data about precarious habitat (vulnerability to natural risks, improvement of habitat ….).

Recommendation 8 A construction that is considered as permanent enough to be captured in existing data should be published for INSPIRE theme *Buildings*, especially if the construction hosts human activities.

NOTE 2: All buildings, whatever their size is, are in the scope of theme *Buildings*. As explained in clause 1.2.2, the scope is modular with first priority to the big or normal buildings. However, there are exceptions where small buildings are of great interest, such as a hut in mountainous area that may be a valuable landmark or shelter for hikers. Once again, this is generally already taken into account by the capture rules of data producers.

NOTE 3: The construction must be above or underground, i.e. it must have a significant height. This excludes "flat" constructions such as roads, railways that should be reported in INSPIRE theme Transport. On the opposite, constructions that are totally or partly underground (bunker, underground stations, underground car parks, swimming pools…) are under scope of theme Building and should be published for INSPIRE, if data is available.

NOTE 4: The constructions that are not buildings and that are already in another INSPIRE theme should generally not be included in the scope of theme *Buildings*, except if attributes typical to theme *Buildings*, such as height or date of construction are required by use cases.

9.6 Use of Building and BuildingPart

9.6.1 When to split a Building into BuildingParts?

BuildingPart is generally used for buildings that are not homogeneous, regarding attributes related to physical aspects (such as height above or below ground , height below ground, number of floors above or under ground, roof type), temporal aspects (such as year of construction) or functional aspects (building nature or current use). A BuildingPart may be used for a contiguous part of a building of which one or more attributes (except identifier and geometry) differ from all other parts it touches.

EXAMPLE 1:

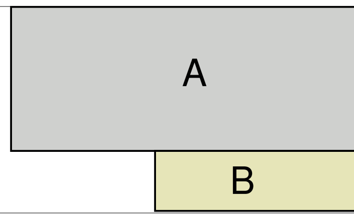
		
Real world building	The Building may be split into 2 BuildingParts A and B because of different height above ground (e.g. 8 m for A, 6m for B)	The building may be represented just as single generalised Building (e.g. with height above ground = 8 m)

Figure 64: Split into building parts (example 1)

EXAMPLE 2:

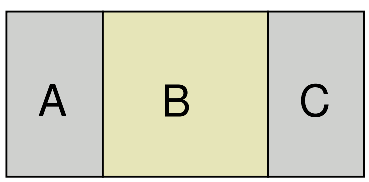
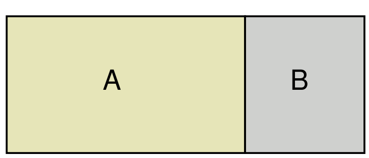
		
Real world building	This Building may be split into 3 BuildingParts A, B and C because of different number of floor above ground (e.g. 20 floors for A and B, 5 floors for B)	The building may be represented just as single generalised Building (e.g. with number of floors above ground = 20)

Figure 65: Split into building parts (example 2)

EXAMPLE 3:

		
Real world building	This Building may be split into 2	The building may be

	BuildingParts A and B because of different current use (agriculture for A, residential for B)	represented just as single generalised Building with current use = {residential, agriculture}

Figure 66: Split into building parts (example 3)

EXAMPLE 4:

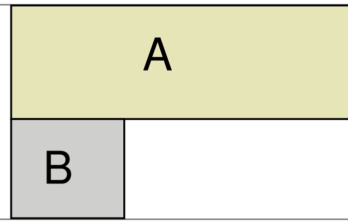
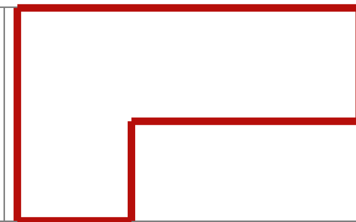
		
Real world building	This Building may be split into 2 BuildingParts A and B because of different date of construction (e.g. 1920 for A, 1950 for B) and roof type (gable roof for A, pavillon roof for B)	The building may be represented just as single generalised Building with date of construction = 1920 (and date of renovation = 1950 if enlargement is considered as renovation)

Figure 67: Split into building parts (example 4)

9.6.2 How to split a Building into BuildingParts?

This data model is quite flexible. It is up to the data capture rules of each data producer to define the relevant building parts. These rules should be explained in the template for additional information.

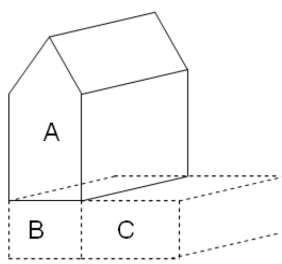


Figure 68: Example from Germany

On the previous figure, the building has been split into 3 building parts, with complete overlap between building parts A and B.

	A	B	C
Number of floors above ground	3	0	0
Number of floors below ground	0	1	1

EXAMPLE 2:

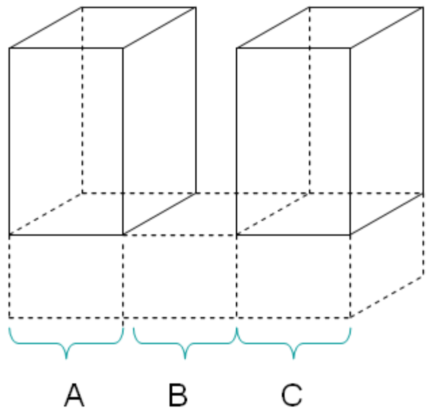
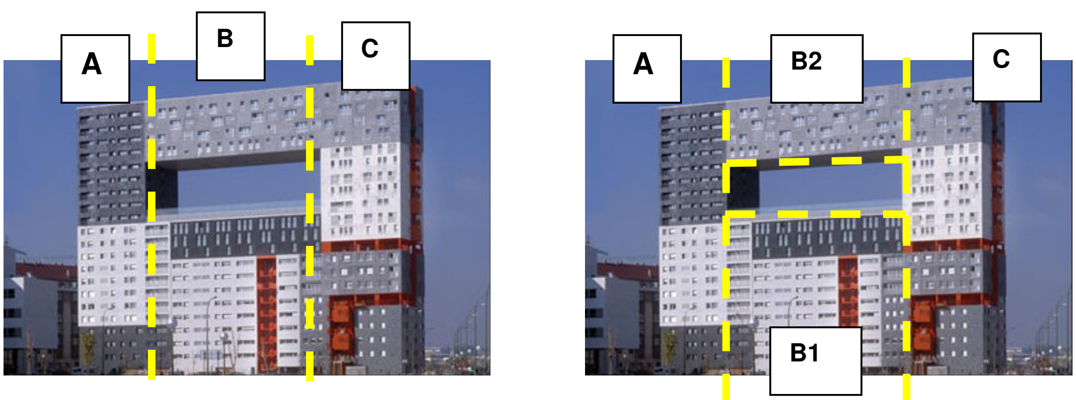


Figure 69: Example from Spain

On the previous figure, the building (two blocks of flats sharing common basement) has been split into 3 non overlapping building parts

	A	**B**	**C**
Number of floors above ground	8	0	8
Number of floors below ground	3	3	3

EXAMPLE 3



This building has been split into 3 building parts

This building has been split into 4 building parts

Figure 70: Building parts with various floor distributions.

	A	B	C
Floor distribution	[0,21]	[0,12], [18, 21]	[0, 21]

	A	B1	B2	C
Floor distribution	[0,21]	[0,12]	[18, 21]	[0, 21]

9.6.3 How to fill the attributes of Building and BuildingPart?

- The mandatory attributes **inspireId** and **geometry** have to be filled on both Building and BuildingPart.

- If available, the attributes **beginLifespanVersion** and **beginLifespanVersion** have also to be filled on both Building and BuildingPart.

- If available, the attributes **numberOfDwellings and numberOfBuildingUnits** may be filled on both Building and BuildingPart with the consistency rules:
  - number of dwelling on Building = sum of number of dwellings of the BuildingParts composing the Building
  - similar rule with numberOfBuildingUnit

- Among the other attributes:
  - The specific attributes shall and can be filled only on Building Parts
  - The common attributes should be filled only on *Buildings*.

9.7 Geometric representation

9.7.1.1. Multiple representation

The INSPIRE model is quite flexible as it allows multiple representations for buildings and building parts. However, not all allowed representations are meaningful and relevant for any kind of buildings.

EXAMPLE:

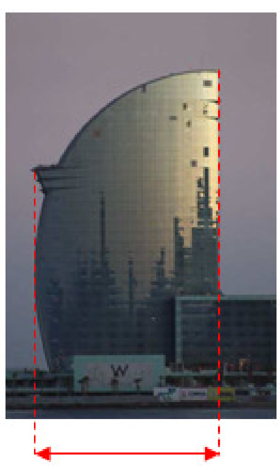
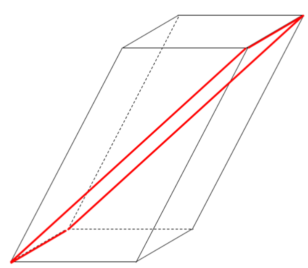
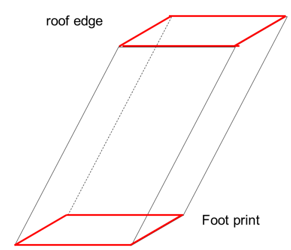
			
	Real-world building	Building represented by above ground envelope	Building represented by foot print and roof edge
Above ground envelope geometry is relevant	The representation by above ground envelope geometry is less relevant than the representation by footprint and roof edge.		

Figure 71: Various geometric representations of buildings

Recommendation 9 Data producers should publish for INSPIRE the most relevant geometric representation(s) of buildings and building parts.

NOTE: The representation of envelope geometries by 2,5D data may raise issues; typically, the Z value may be not very meaningful.

9.7.1.2. Missing Z and 2,5D data

Some existing data sets may include both 2,5 D and 2D data. This is typically the case for data sets where most buildings are captured by stereo-plotting with Z coordinate (i.e. as 2,5D data) whereas some others are captured by other ways (e.g. from existing maps or from orthoimages) without the Z coordinate (i.e. as 2D data).

This case may be avoided if the data producer can wrap the buildings on a DTM and so, can attribute to the building geometric representation a reasonable Z value at ground level; the vertical geometry accuracy enables to document the process approximation.

Nevertheless, for cases where there are still missing Z values on some buildings, the general rule is to attach the Coordinate Reference System at feature level (instead of declaring it for the whole data set):
- *Buildings* captured as 2,5D data will be attached with a 3D Coordinate Reference System
- Buldings captured as 2D data will be attached with a 2D Coordinate Reference System

Normally, the 3D Coordinate Reference System will be a compound system whose 2D component is the same Coordinate Reference System as the one used for the buildings captured as 2D data.

An alternative solution would be to provide 2D and 2,5D buildings in different data sets.

9.8 Mapping examples for attribute currentUse

The principle is to match at the most detailed level as possible. Some approximate mappings are acceptable and even necessary. However, they should be reported in the template for additional information (Annex E)

Example 1: from Dutch Dwelling Register to INSPIRE

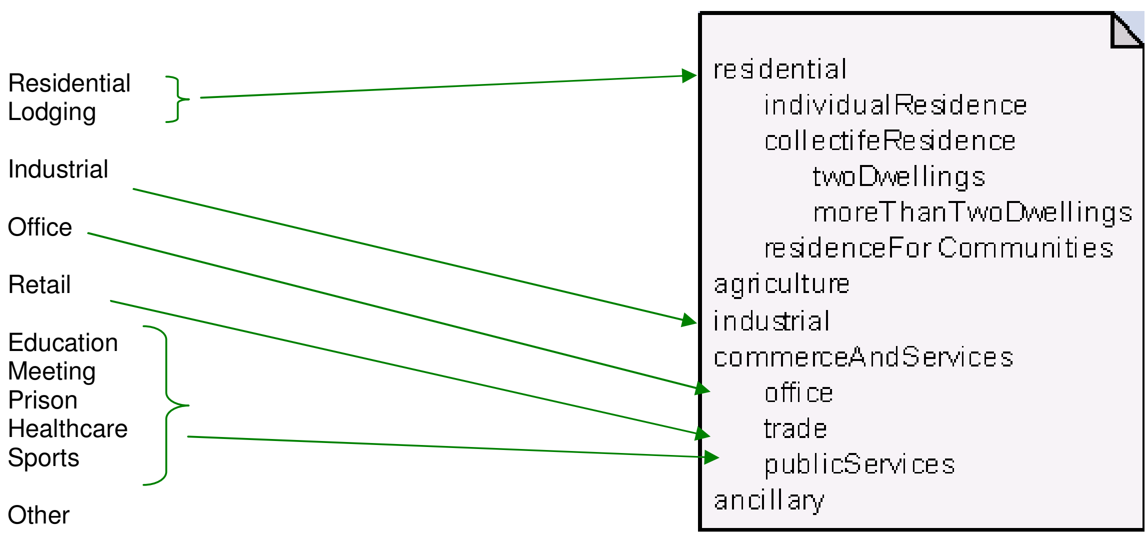


Figure 72: Matching example from national classification to INSPIRE classification of current use

Example 2 : from Eurostat classification to INSPIRE

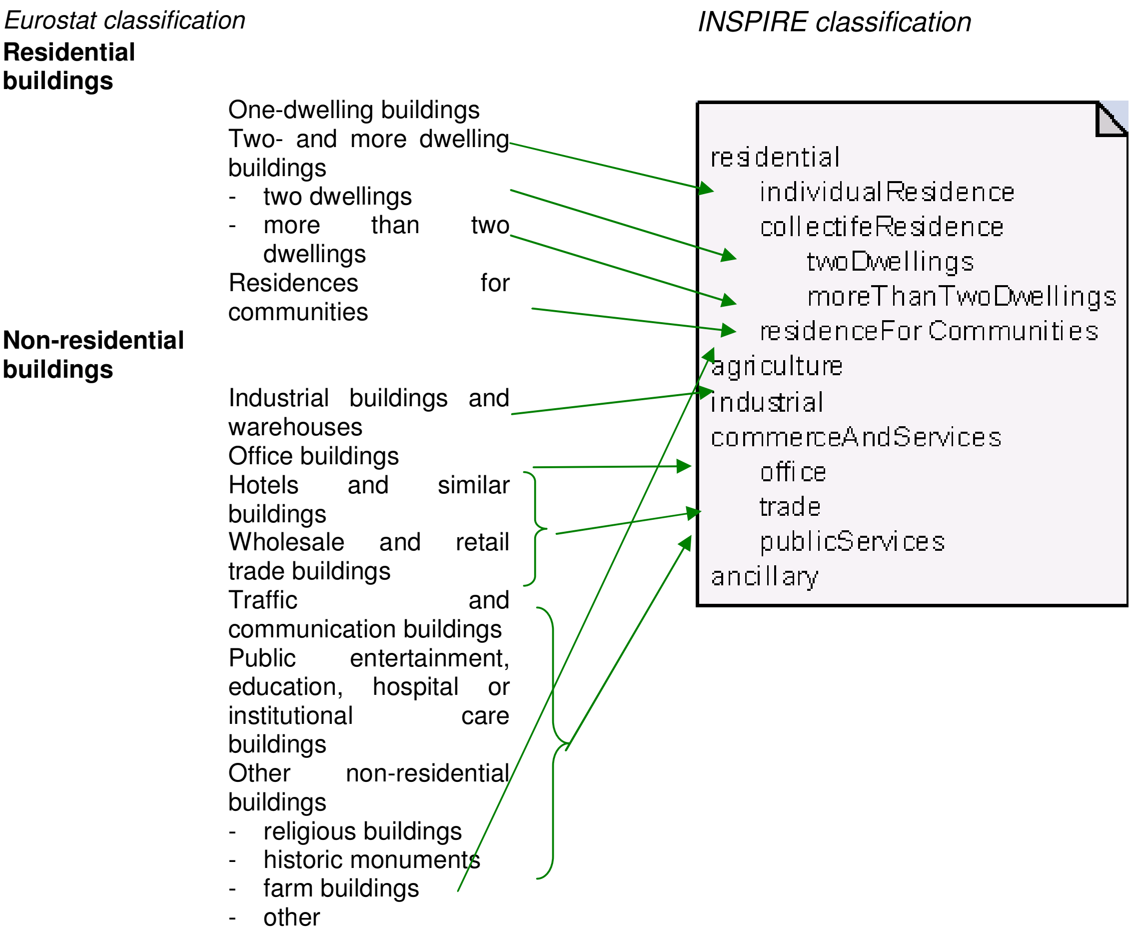


Figure 73: Matching example from EUROSTAT classification to INSPIRE classification of current use

NOTE 1: some data producers have already implemented the Eurostat classification.

9.9 Temporal aspects

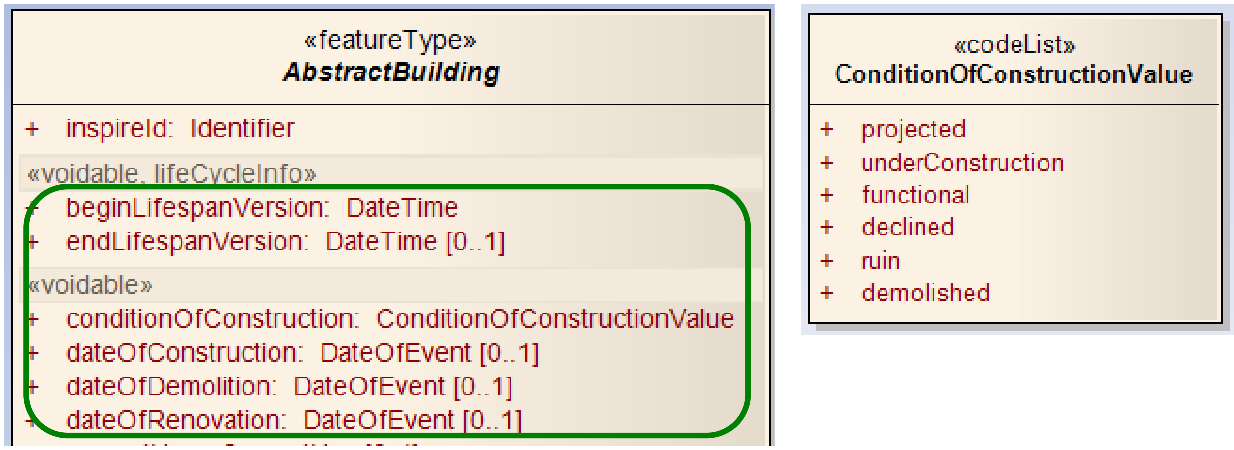


Figure 74: Temporal aspects

The INSPIRE UML schema includes 6 attributes that are related to the temporal aspects:
- conditionOfConstruction: current condition of the construction or building
- date of construction, date of renovation and date of demolition that are related to respective events in the real world
- beginLifespanVersion and endLifespanVersion that are related to the events in the dataset (e.g. when a construction was inserted in the data set or when it was depreciated).

9.9.1 Data type DateOfEvent

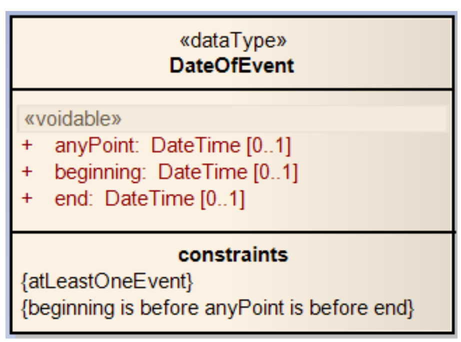


Figure 75: Data type DateOfEvent

The data type DateOfEvent enables to supply temporal information about an event (construction, renovation, demolition) in the following cases:
- a data producer has the date of the event but without any other information about which phase of the event the date refers to
- a data producer does not have the date of the event but has the information as an interval (e.g. before 1950, between 1800 and 1900); this case applies mainly for old buildings
- a data producer has several dates corresponding to different points of the event, e.g. the beginning and the end of the event.

EXAMPLES (for date of construction)
- producer knows that construction date is 1978
  - beginning: void
  - end: void

  - anyPoint: 1978
- producer knows that construction took place before 1950
  - beginning: void
  - end: 1950
  - anyPoint: void
- producer knows that construction took place between 1800 and 1900
  - beginning: 1800
  - end: 1900
  - anyPoint: void
- producer knows that construction took place between 12/04/2008 and 25/12/2010
  - beginning: 12/04/2008
  - end :25/12/2010
  - anyPoint: void

9.9.2 Demolished Buildings

There are two ways to deal with demolished constructions or buildings.

EXAMPLE: a building that was functional was demolished on 20/03/2010 and this information is integrated by data producer on the 15/05/2010

- first method: the building is considered as depreciated (no valid any longer)
  - its attribute endLifespanVersion gets value “15/05/2010”
  - its attribute dateOfDemolition gets value “20/03/2010”
  - the other attributes stay as they are, describing the building as it was just before being demolished (e.g. its attribute conditionOfConstruction remains “functional”)

- second method: the building is versioned in the database :
  - the attribute endLifespanVersion of the old version of the building will get value “15/05/2010”
  - the attribute dateOfDemolition of the old version remains empty
  - the other attributes stay as they are, describing the building as it was just before being demolished (e.g. its attribute conditionOfConstruction remains “functional”)

  - the attribute beginLifespanVersion of the new version of the building will get value “15/05/2010”
  - the attribute endLifespanVersion of the new version of the building remains empty
  - the attribute conditionOfConstruction of the new version will get value “demolished”
  - the attribute dateOfDemolition of the new version will get value "20/03/2010"

The second method is well adapted for the data sets that aim to manage historical buildings whereas the first one is probably better for data sets that aim to manage current buildings.

9.10 Estimated accuracy

For INSPIRE, buildings shall be published in the Coordinate Reference System mandated by the Implementing Rule on Reference Systems, i.e. in ETRS89 for areas on the Eurasian tectonic plate and in ITRS elsewhere.

Of course, INSPIRE users will be interested by having information about the accuracy of building data, as they receive them, in the Coordinate Reference System mandated by INSPIRE. It is why the clauses about application schema and about quality and metadata require building data providers to give estimated accuracy related to the coordinates in ETRS89 (or ITRS).

However, in most Member States, the estimated accuracy is generally known in the source Coordinate Reference System, the national or local one.

The estimated accuracy for INSPIRE will be the combination of estimated accuracy in original Coordinate Reference System and of the accuracy of the coordinate transformation between original Reference System to INSPIRE Reference System.

Coordinate transformation between two horizontal geodetic datum is generally done, using one of the three following methods:

- transformation with 3 parameters
- transformation with 7 parameters
- transformation with a grid.

Experience in some countries has shown that transformation with 3 or even 7 parameters might bring deviations up to 10 metres. So, the impact of such transformations may not be neglected on building data whose original accuracy generally varies from some decimetres to some metres.

The ideal solution would be for each Member State to define good quality coordinate transformations (using grids and bringing no deviation bigger than some centimetres). However, if not possible before the deadlines of INSPIRE, the impact of coordinate transformation has to be taken into account when giving information about positional accuracy, both in the application schema and in metadata.