INSPIRE

Infrastructure for Spatial Information in Europe

Proposed Changes to the Generic Conceptual Model and Encoding Guidelines

Title	Proposed Changes to the Generic Conceptual Model and Encoding Guidelines
Creator	CPO
Date	2011-06-15
Subject	Changes to the conceptual modelling framework of the INSPIRE data specifications as proposed by version 2.0 of data specifications for the spatial data themes of Annex II/III
Publisher	INSPIRE Drafting Team "Data Specifications"
Type	Text
Description	The requirements and recommendations specified in the document have been identified in the cross-theme-discussions during the process of developing the data specifications for Annex II and III. Based on the results of the consultation and testing, these items may be revised and moved into a new version of the Generic Conceptual Model and the Guidelines for the encoding of spatial data. Other topics identified during this process may also be addressed in that revision.
Contributor	JRC DS Team, DT DS, TWGs
Format	MS Word (doc)
Source	
Rights	Public comments
Identifier	ChangesToD2.5andD2.7.docx
Language	En
Relation	n/a
Coverage	Project duration

Table of contents

1 Scope

This document specifies amendments to the Generic Conceptual Model for INSPIRE (D2.5), version 3.3, and the Guidelines for the encoding of spatial data (D2.7), version 3.2, as required by the candidate INSPIRE data specifications version 2.0 for the spatial data themes of Annex II/III.

NOTE 1 The following formatting conventions are used in this document.

Explanatory text is shown highlighted with a yellow background.

Open issues are shown highlighted with a light red background.

NOTE 2 New requirements and recommendations use N1, N2, etc as identifiers to distinguish them from those in D2.5 version 3.3 and D2.7 version 3.2.

2 Additional normative references

D2.9 Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development, version 1.0

OGC 09-146r1, GML application schema – Coverages, version 1.0

3 Additional terms and abbreviations

3.1 Terms

(1) code list
open **enumeration** that can be extended [ISDSS Regulation]

NOTE The definition supersedes the current definition, which has been "value domain including a code for each permissible value"

(2) continuous coverage
coverage that returns different values for the same **feature attribute** at different **direct positions** within a single geometry object in space and/or time in its **domain** [adapted from ISO 19123]

(3) direct position
position described by a single set of coordinates within a **coordinate reference system** [ISO 19107]

(4) discrete coverage
coverage that returns the same **feature attribute** values for every **direct position** within any single geometry object in space and/or time in its **domain** [adapted from ISO 19123]

(5) domain
well-defined set [ISO/TS 19103]

(6) enumeration
data type whose instances form a fixed list of named literal values [ISDSS Regulation]

NOTE The definition supersedes the current definition, which has been "data type whose values are enumeration literals "

(7) feature attribute
characteristic of a **feature** [ISO 19101]

(8) function
rule that associates each element from a **domain** (source or domain of the function) to a unique element in another domain (target, co-domain or **range**) [ISO 19107]

(9) range <coverage>
set of **feature attribute** values associated by a **function** with the elements of the **domain** of a **coverage** [ISO 19123]

3.2 Abbreviations

O&M	OGC Observation and Measurement
WCS	OGC Web Coverage Service

4 Overview

The amendments are related to the following topics, which are detailed in the subsequent clauses:

- Use of code lists
- Use of coverage functions
- References to observations from spatial objects
- Additional data types used by several themes
- Clarifications on references to spatial objects, metadata elements and the SpatialDataSet type

5 Use of code lists

The requirements and recommendations listed in the following sections reflect the needs identified during the development of the Annex II&III data specifications. They are meant to replace the requirements and recommendations contained in D2.5 version 3.3 and D2.7 version 3.2 (see Annex A).

5.1 Enumerations vs. code lists

Recommendation N1 In the case of an attribute type with coded values, an enumeration or code list should be used. If the set of allowed values is fixed, an *enumeration* should be used. If the set of allowed values may be extended by user communities or without a major revision of the data specification, a code list should be used.

Open issue 1: It is still an open issue, whether a distinction shall be made between code lists with and without internal structure.

It has been proposed to refer to simple lists (or more precisely, sets) of values as *code lists*, and to collections of values that have some internal (e.g. a hierarchical) structure as *concept schemes*. However, it has been agreed between the Annex II+III TWGs *not* to make this distinction in v2.0 of the INSPIRE data specifications, but to refer to both kinds of value domain as *code lists* – even if, conceptually, this is not entirely correct.

The issue will be discussed in the relevant ISO TC 211 committees (notably on the revision of ISO 19103 and ISO 19109), and any decision taken there will be adopted by INSPIRE.

EXAMPLE 1 To describe the days of the week, the following values are distinguished:
- monday
- tuesday
- wednesday
- thursday
- friday
- saturday
- sunday

As this list of possible values may be considered as exhaustive, it is not likely to change. Therefore, it should be specified as an **enumeration** in the application schema.

EXAMPLE 2 To describe the hydrological persistence of a body of water, the following values are currently in use:
- dry
- ephemeral
- intermittent
- perennial

As this list of possible values may be considered as non-exhaustive (e.g. the user requirements may evolve and this list may have to be enriched), it should be specified as a **code list** in the application schema.

5.2 Extendable vs. non-extendable code lists

Requirement N1	Code lists shall be of one of the following types: (a) code lists that shall not be extended by Member States; (b) code lists that may be extended by Member States.

EXAMPLE 1 The values included in the "hydrological persistence" code list shown in EXAMPLE 2 should be the same for all applications, this code list would be centrally managed.

EXAMPLE 2 In contrast, the code list containing the allowed values for designation schemes for protected sites (currently containing international designation schemes such as Natura 2000 or Ramsar) should be extensible so that also national designation schemes can be used.

NOTE For instructions on how code lists can be extended by Member States, see section 5.7.

5.3 External code lists

In many Annex II/III themes, there are existing code lists that could be re-used in INSPIRE application schemas.

Recommendation N2	If for a specific property in an INSPIRE data specification, there is already an existing code list that is maintained by an international organisation and meets the requirements set out in Requirement N2, this code list should be referenced from the INSPIRE application schema.

Requirement N2	An externally managed code list may only be referenced from an INSPIRE application schema if it meets the following requirements: 1. It has to be managed by a competent international organisation. 2. It must be well maintained, i.e. all its values must remain available forever, even if they have been deprecated, retired or superseded. 3. The code list and each of its values must be identifiable through a persistent URI1 in the 'http' scheme. Exceptions are values of code lists that are only available as HTML. 4. The code list must be available in HTML plus at least one of the following machine-readable representations: - GML dictionary - SKOS For a transition period, also code lists that are only available in HTML, may be used in INSPIRE. In this case, it shall be clearly specified in the UML application schema how to identify (in the HTML representation) the values' unique identifiers that shall be used.

NOTE 1 The adoption of a third-party register creates a dependency and requires a discussion between INSPIRE and the organisation managing the external register. This includes intellectual property rights aspects, notification of changes, etc. The outcome of this discussion could be documented in an (informal) agreement between the external organisation and the Commission / Member States.

Open issue 2: It has to be clarified whether/how references to external lists are permissible in a Regulation and how dependencies should be formally agreed and/or documented.

[1] In INSPIRE, Uniform Resource Identifiers (URIs) as defined in IETF RFC 3986 are intended to serve as these persistent, location-independent resource identifiers.

Requirement N3	If an external code list should be used in an INSPIRE application schema, but does not meet the requirements set out in Requirement N2, the values shall be made available through the common INSPIRE code list registry.

NOTE 2 This representation has to be maintained and updated if the reference in the INSPIRE application schema is updated to a new version of the external code list. A workflow for this should be defined as part of the terms of reference for maintenance and implementation.

EXAMPLE 1 The 4 values of the AerodromeTypeValue code list in the Air Transport Network application schema was derived from AIXM5.0.

EXAMPLE 2 The values of the SpeciesTypeValue code list in the draft Species Distribution application schema (created by the TWG PS during the Annex I development) is based on the species lists from Article 4.1 and 4.2 of the Birds Directive (Council Directive 79/409/EEC) and Annex II of the Habitats Directive (Council Directive 92/43/EEC).

5.4 Code lists in UML application schemas

5.4.1 Stereotypes

Requirement N4	Code lists shall be included as classes with the stereotype <<codeList>> in the UML application schemas.

5.4.2 Tagged values

Requirement N5	If one or one of several specific code lists is to be mandated in the application schema, a tagged value “codeList” shall be provided for each code list that may be used. The value shall be a persistent URI identifying the code list. If no tagged value “codeList” is provided, then an arbitrary code list compliant with Requirement N2 and the definition of the code list in the application schema may be used in conformant data.

EXAMPLE 1 The definition of the *CountryCode* code list in the *Base Types* application schema of the Generic Conceptual Model refers to the values defined in the Interinstitutional style guide published by the Publications Office of the European Union (http://publications.europa.eu/code/en/en-370100.htm). According to the requirements and recommendations in this section, the application schema should include the “codeList” tagged value, the code list should be well maintained and it should be clearly defined how, in the HTML representation, one can identify the values’ unique identifiers that shall be used.

NOTE For code lists that are to be centrally managed in the INSPIRE code list register, the value of the tagged value “codeList” will be preliminarily set to http://inspire-registry.jrc.ec.europa.eu/clr/<name of the class>. This value may change once the INSPIRE code list register has been established.

Requirement N6	To distinguish the different types of code lists, a tagged value "extendableByMS" shall be provided for each code list. The value shall be "false" for code lists that may not be extended by Member States (type (a) in Requirement N1) and "true" for code lists that may be extended (type (b) in Requirement N1).

5.4.3 Class names

Recommendation N3	The class name for enumerations and code lists should include the suffix "Value".

EXAMPLE 1 IndustryTypeValue, SoilTypeValue

5.4.4 Versioned vs. unversioned references

If one or several external code lists are referenced from the application schema through tagged values, this reference can be:

a. to a specific version (identified by a date or version number) of the code list.
b. to the code list in general – without referring to a specific version.

Recommendation N4	Where possible, the reference to the code list in the "codeList" / "conceptScheme" tagged value(s) should not be to a specific version.

Open issue 3: It has to be clarified whether this approach is permissible in a Regulation.

5.4.5 References to subsets of external lists

Recommendation N5	To reference only a subset of an externally managed code list (e.g. only the top level(s) of a hierarchical code list or a certain thematic subset), a constraint should be placed on the code list class specifying which subset of the external code list shall be used.

NOTE For pragmatic reasons, the subset could additionally be made available through the common INSPIRE code list registry.

EXAMPLE 1 Different subsets of the NACE codes (http://ec.europa.eu/competition/mergers/cases/index/nace_all.html) may be relevant for different themes, e.g. *Section A: Agriculture, hunting and forestry* for AF, *section C: Mining and quarrying* for MR, *sub-section CA: Mining and quarrying of energy producing materials* for PF and ER and *section O: Other community, social and personal service activities* for US.

5.5 Values of code lists

Requirement N7	The values of code lists shall be managed only in registers outside of the application schema.

NOTE 1 The notion "register" includes the IR-ISDSS, which factually is the common INSPIRE code list register.

Recommendation N6	The values of code lists should still be included in the UML application schemas for illustration and to allow constraints to be included in the application schema that refer to code list values. Where it is not feasible to include all code list values (e.g. because there are too many), this should be mentioned in a note.

Requirement N8	If a code list does not contain any values, it shall be declared as an abstract class.

NOTE 2 Requirement N7 requires that the values of code lists be managed in a register. Since no such register is currently available for code lists, the values are currently only included in the UML model. For hierarchical code lists, it is proposed as an interim solution to prefix the values with numbers divided by underscores to indicate different levels.

EXAMPLE 1 The values of an energy commodity code list could be structured as follows:

1_crudeOil
1_1_bitumen
...
2_naruralGas
2_1_acquiferGas
...

Requirement N9	Values of code lists and enumerations shall be all in lowerCamelCase notation: - They shall not contain any whitespace. - Only the first letter of each word after the first word that is combined in the value name is capitalised. Exceptions are words that consist of all uppercase letters (acronyms) and values of externally managed code lists.

Requirement N10	The values of code lists shall be made available through a registry. All instances have to reference the registry to allow applications to evaluate the code list value and its validity, its title in the official languages or its definition.

Recommendation N7 URIs should be used to encode identifiers of items in INSPIRE code list registers and to reference such items.

The URIs should use the following structure:

http://inspire-registry.jrc.ec.europa.eu/<register>/<item class>/<item identifier>

where
- <register> is the name of the register (ISO 19135: RE_Register);
- <item class> is the name of the item class (ISO 19135: RE_ItemClass) of the registered item;
- <item identifier> is the item identifier of the registered item (ISO 19135: RE_RegisterItem).

Colons should not be used in <item class>, <register> or <item identifier> values.

See also Issue 1 in D2.7, for now this is a preliminary URI pattern and implemented at this time.

Other URIs may be used, too, for items managed in external registers. For registers with stable URIs as identifiers, these URIs may be used instead. For external registers that do not provide stable URIs, the “inspire”-URIs will be used, but the <register> will identify the external register and authority.

NOTE 3 The use of HTTP includes support for content negotiation to access a resource in different languages and/or different encodings, if supported. The approach also enables universal access and understanding by a wider community. The use of other URIs is intended to support the usage of URIs that are already commonly used and supported by software components or specified in other standards.

EXAMPLE 2 http://inspire-registry.jrc.ec.europa.eu/clr/125 would be a reference to a code list in the INSPIRE code list register. The item identifier 125 is unique and identifies the item within the register. When an item is superseded by a new version, it receives a new identifier, and the superseded item is retired.

NOTE 4 The name of the code list cannot be used as the item identifier as over time the same name may be associated with different concepts; e.g., when the definition of a code list is amended.

EXAMPLE 3 We propose to have a hierarchical register of code lists, where each code list can be seen as a separate register (of values). Thus, to refer to a value (item class: value) of this code list, the identifier could be: http://inspire-registry.jrc.ec.europa.eu/clr/125. The name of the register would then be: “clr/125”.

5.6 Using values in instance data

Requirement N11	Attributes of spatial object types or data types whose type is a code list that may not be extended by Member States (type (a) in Requirement N1) may only take values included in (one of) the code list(s) specified in the “codeList” tagged value(s).

	Attributes of spatial object types or data types whose type is a code list that may be extended by Member States (type (b) in Requirement N1) may only take values that are valid according to the register in which the code list is managed.

Requirement N12	When specifying a code list value in instance data, a reference shall be given to the unique identifier of the value.

Recommendation N8	When specifying a code list value in instance data, a human-readable label to be used in user interfaces should be provided in addition to the unique identifier of the value.

EXAMPLE 1 In GML, the value identifier and label for the country code (code list id: 123) for Germany (value id: 15) could be encoded using the xlink:href and xlink:title properties:

```
<SomeFeature gml:id="abc123">
  <country xlink:href="http://inspire-registry.jrc.ec.europa.eu/clr/123/15" xlink:title="Deutschland"/>
  (…)
</SomeFeature>
```

5.7 Member State extensions of code lists

Requirement N13	Where a Member State extends a code list, the allowed values of the extended code lists shall be made available in a register.

NOTE Member states can only extend code lists defined in defined in Requirement N1b. How national code list registers are set up, is an issue that should be discussed in the working group currently under discussion for INSPIRE maintenance and implementation.

EXAMPLE 1 The *DesignationSchemaValue* from the *Protected Sites Simple* application schema (shown in orange) contains values for a number of international and European designation schemas:

- emeraldNetwork
- IUCN
- nationalMonumentRecord

- natura2000
- ramsar
- UNESCOManAndBiospereProgramme
- UNESCOWorldHeritage

In this example, this list is extended with a national designation schema, the designation schema according to the German nature protection act:
- BNatSchG

The MS application schema also includes an additional code list (*BNatSchGDesignationValue*) that contains the designation values for the new designation scheme. This code list generalises the abstract empty code list *DesignationValue* defined in the *Protected Sites Simple* application schema:

- biosphaerenReservat
- biotope
- geschuetzterLandschaftsbestandteil
- landschaftsSchutzGebiet
- nationalParkNationalesNaturMonument
- naturDenkmal
- naturPark
- naturSchutzGebiet

6 Use of coverage functions

The data specifications for the Annex I themes did not include any coverage functions in their application schemas. This changes with Annex II/III. For the sake of harmonisation and considering the availability of implementations supporting coverages, the guidelines were refined as described in this clause. In particular, the new GML application schema for coverages specified as part of the OGC Web Coverage Service standard version 2.0 (OGC document 09-146r1) is used as a basis for representing coverage information to simplify the use of coverages in standards-based download services.

This model has been amended to meet additional INSPIRE requirements.

The current model should be understood as capturing coverage-related requirements of INSPIRE, but the eventual guidance and implementation may differ as long as it meets the requirements and is as far as possible is based on standards. SDICs and LMOs are therefore invited to submit comments on these aspects during the testing and consultation phase.

6.1 Rules for use of coverage functions

This sub-clause replaces sub-clause 10.4 of D2.5 version 3.3.

Coverage functions are used to describe characteristics of real-world phenomena that vary over space and/or time. Typical examples are temperature, elevation, precipitation, imagery. A coverage contains a set of such values, each associated with one of the elements in a spatial, temporal or spatio-temporal domain. Typical spatial domains are point sets (e.g. sensor locations), curve sets (e.g. contour lines), grids (e.g. orthoimages, elevation models), etc. A continuous coverage is associated with a method for interpolating values at spatial positions between the elements of a domain, e.g. between two points or contour lines.

Rules on coverage functions are not part of the current version of ISO 19109 since that standard predates ISO 19123. Therefore, the following rules are added in the Generic Conceptual Model, based on the extension to General Feature Model specified in D2.5, version 3.3, 9.2.3.

Requirement N14	Any specification of a coverage function in an INSPIRE application schema shall be in accordance with ISO 19123 and D2.5, version 3.3, 9.2.3.

Requirement N15	An application schema package that uses coverage functions shall follow the rules of ISO 19109 8.2.5 for referencing standardized schemas, i.e. import directly or indirectly the coverage schema specified by ISO 19123.

Requirement N16	A coverage function shall be defined as a property of a spatial object type where the type of the property value is a realisation of one of the types given in Table 1.

NOTE 1 This approach creates an implementation profile of the relevant ISO 19123 types in the INSPIRE application schema (or the Generic Conceptual Model). See also 6.2.

Table 1 - List of valid coverage types in an application schema

Abstract coverage types	**Discrete coverages**	**Continuous coverages**
CV_Coverage CV_DiscreteCoverage CV_ContinuousCoverage	CV_DiscretePointCoverage CV_DiscreteGridPointCoverage CV_DiscreteCurveCoverage CV_DiscreteSurfaceCoverage CV_DiscreteSolidCoverage	CV_ThiessenPolygonCoverage CV_ContinousQuadrilateralGridCoverage CV_HexagonalGridCoverage CV_TINCoverage CV_SegmentedCurveCoverage

NOTE 2 The implementation of ISO 19123 within ISO 19136 / GML and the OGC Web Coverage Service is incomplete from the point of view of some of the themes. For example, spatio-temporal coverage domains, the continuous coverage types, and the irregular nature of some atmospheric and oceanic model coverages (e.g. using CV_ReferenceableGrid) are insufficiently specified. Change requests are being discussed and/or processed to address two of these issues:

- Two OGC Change Requests (07-112r3 and 09-091r3) add encodings for three implementation variants of CV_ReferencableGrid (by array, by vectors and by transformation) and currently form part of the GML 3.3 draft.
- Beside the range/domain representation, requirements for geometry/value pair representations of coverages exist. This is addressed by the OGC best practice document 06-188r1 which provides a schema for this "interleaved pattern" as it is used, e.g., for a time series. This requires an amendment to the coverage schema used by WCS 2.0.

NOTE The use of the Web Coverage Service as an INSPIRE download service is a separate issue outside of the scope of this document.

6.2 Profile of the coverage schema

This sub-clause replaces sub-clause 10.5 of D2.5 version 3.3.

Coverage types as specified in ISO 19123 and in 6.1 may be used in an INSPIRE application schema. To improve alignment with coverage standards on the implementation level and to improve the cross-theme harmonisation on the use of coverages in INSPIRE an application schema for coverage types is included in the Generic Conceptual Model in 9.9.4.

Recommendation N9	Coverage types used in INSPIRE application schemas should be subtypes of the coverage types specified in 9.9.4.

Table 2 - List of recommended coverage types in an application schema

Abstract coverage types	Discrete coverages by domain/range	Continuous coverages by geometry/value pairs
Coverage DiscreteCoverage ContinuousCoverage CoverageByDomainAndRange	MultiPointCoverage MultiCurveCoverage MultiSurfaceCoverage MultiSolidCoverage GridCoverage RectifiedGridCoverage ReferenceableGridCoverage	DiscreteCoverageGeometryValuePairs MultiPointCoverage MultiCurveCoverage MultiSurfaceCoverage MultiSolidCoverage MultiTimeInstantCoverage

6.3 Coverages

This sub-clause adds a new sub-clause 9.9.4 to D2.5 version 3.3.

6.3.1 Feature Catalogue - Coverages (Base)

Table 3 - Feature catalogue metadata

Feature catalogue name	INSPIRE feature catalogue Coverages (Base)
Scope	Coverages (Base)
Version number	1.0
Version date	2011-06-01
Definition source	INSPIRE data specification Coverages (Base)

Table 4 - Types defined in the feature catalogue

Type	Package	Stereotypes	Section
Coverage	Coverages (Base)	«featureType»	6.3.1.1.1

6.3.1.1 Spatial object types

6.3.1.1.1 Coverage

Coverage (abstract)	
Definition:	feature that acts as a function to return values from its range for any direct position within its spatial, temporal or spatiotemporal domain
Description:	EXAMPLE Examples include a raster image, polygon overlay or digital elevation matrix. NOTE In other words, a coverage is a feature that has multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Attribute: metadata	
Value type:	Any
Definition:	application specific metadata of the coverage
Description:	NOTE The values of this property will typically be constrained in subtypes or in profiles specified by information communities.
Multiplicity:	0..*
Attribute: rangeType	
Value type:	RecordType

Coverage (abstract)	
Definition:	description of the structure of the range values
Multiplicity:	1

6.3.1.2 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.

6.3.1.2.1 Any

Any	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19103 Conceptual Schema Language::ISO 19103:2005 Schema Language::Basic Types::Implementation::Records and Class Metadata

6.3.1.2.2 RecordType

RecordType	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19103 Conceptual Schema Language::ISO 19103:2005 Schema Language::Basic Types::Implementation::Records and Class Metadata

6.3.2 Feature catalogue – Coverage (Domain and Range)

Table 5 - Feature catalogue metadata

Feature catalogue name	INSPIRE feature catalogue Coverages (Domain and Range)
Scope	Coverages (Domain and Range)
Version number	1.0
Version date	2011-06-01
Definition source	INSPIRE data specification Coverages (Domain and Range)

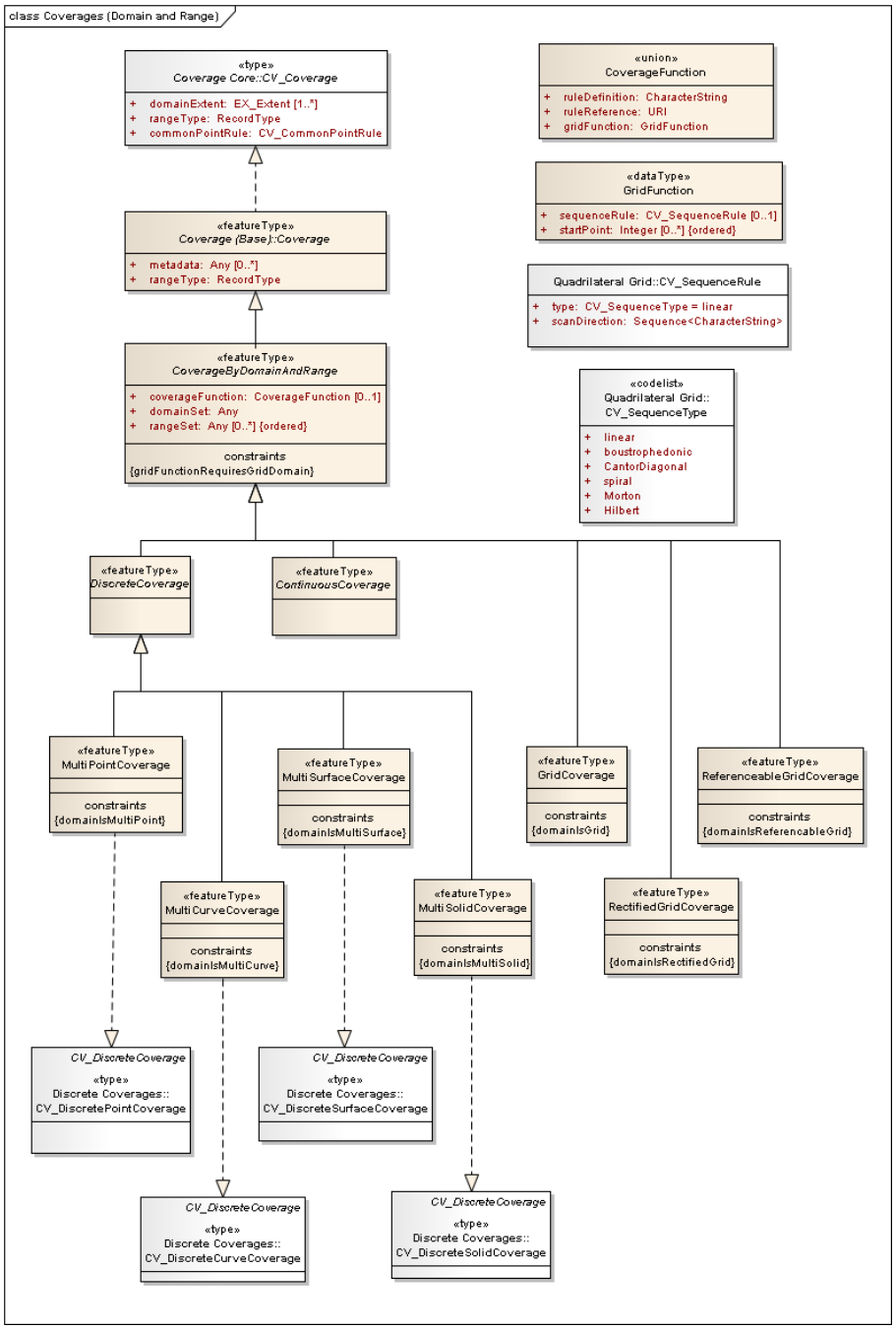

Figure 1 – Coverage representation using a domain/range pair

Table 6 - Types defined in the feature catalogue

Type	Package	Stereotypes	Section
ContinuousCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.1
CoverageByDomainAndRange	Coverages (Domain and Range)	«featureType»	6.3.2.1.2
CoverageFunction	Coverages (Domain and Range)	«union»	6.3.2.2.1
DiscreteCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.3
GridCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.4
GridFunction	Coverages (Domain and Range)	«dataType»	6.3.2.2.2
MultiCurveCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.5
MultiPointCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.6
MultiSolidCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.7
MultiSurfaceCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.8
RectifiedGridCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.9
ReferenceableGridCoverage	Coverages (Domain and Range)	«featureType»	6.3.2.1.10

6.3.2.1 Spatial object types

6.3.2.1.1 ContinuousCoverage

ContinuousCoverage (abstract)	
Subtype of:	CoverageByDomainAndRange
Definition:	coverage that returns different values for the same feature attribute at different direct positions within a single spatial object, temporal object or spatiotemporal object in its domain
Description:	NOTE Although the domain of a continuous coverage is ordinarily bounded in terms of its spatial and/or temporal extent, it can be subdivided into an infinite number of direct positions.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null

6.3.2.1.2 CoverageByDomainAndRange

CoverageByDomainAndRange (abstract)	
Subtype of:	Coverage
Definition:	coverage which provide the domain and range as separate properties
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Attribute: coverageFunction	
Value type:	CoverageFunction
Definition:	description how range values at locations in the coverage domain can be obtained
Multiplicity:	0..1
Attribute: domainSet	
Value type:	Any
Definition:	configuration of the domain of the coverage described in terms of coordinates

CoverageByDomainAndRange (abstract)	
Description:	NOTE The values of this property will typically be constrained in subtypes to specific spatial and/or temporal geometries.
Multiplicity:	1
Attribute: rangeSet	
Value type:	Any
Definition:	set of feature attribute values associated by a function with the elements of the domain of the coverage
Multiplicity:	0..*
Collection Constraints:	ordered
Constraint: gridFunctionRequiresGridDomain	
Natural language:	grid function only valid for domains that are grids
OCL:	inv: coverageFunction.gridFunction.notEmpty() implies domainSet.oclIsKindOf(CV_Grid)

6.3.2.1.3 DiscreteCoverage

DiscreteCoverage (abstract)	
Subtype of:	CoverageByDomainAndRange
Definition:	coverage that returns the same feature attribute values for every direct position within any single spatial object, temporal object or spatiotemporal object in its domain
Description:	NOTE The domain of a discrete coverage consists of a finite set of spatial, temporal, or spatiotemporal objects.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null

6.3.2.1.4 GridCoverage

GridCoverage	
Subtype of:	CoverageByDomainAndRange
Definition:	coverage whose domain consists of a collection of grid points
Description:	NOTE 1 This type can be used for both discrete and continuous coverages. NOTE 2 Some ambiuities exist in the use of CV_Grid as the domain instead of the subtypes of CV_Grid. It is therefore recommended to use RectifiedGridCoverage or ReferenceableGridCoverage for all gridded datasets.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: domainIsGrid	
Natural language:	domain is a grid
OCL:	inv: domainSet.oclIsKindOf(CV_Grid)

6.3.2.1.5 MultiCurveCoverage

MultiCurveCoverage	
Subtype of:	DiscreteCoverage

MultiCurveCoverage	
Definition:	coverage characterized by a finite spatial domain consisting of curves.
Description:	NOTE Often the curves represent features such as roads, railroads or streams. They may be elements of a network. EXAMPLE A coverage that assigns a route number, a name, a pavement width and a pavement material type to each segment of a road system.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: domainIsMultiCurve	
Natural language:	domain is a multi curve
OCL:	inv: domainSet.oclIsKindOf(GM_MultiCurve)

6.3.2.1.6 MultiPointCoverage

MultiPointCoverage	
Subtype of:	DiscreteCoverage
Definition:	coverage characterized by a finite domain consisting of points
Description:	NOTE Generally, the domain is a set of irregularly distributed points. However, the principal use of discrete point coverages is to provide a basis for continuous coverage functions, where the evaluation of the continuous coverage function is accomplished by interpolation between the points of the discrete point coverage. Most interpolation algorithms depend upon a structured pattern of spatial relationships between the points. This requires either that the points in the spatial domain of the discrete point coverage be arranged in a regular way, or that the spatial domain of the continuous coverage be partitioned in a regular way in relation to the points of the discrete point coverage. EXAMPLE A set of hydrographic soundings is a discrete point coverage.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: domainIsMultiPoint	
Natural language:	domain is a multi point
OCL:	inv: domainSet.oclIsKindOf(GM_MultiPoint)

6.3.2.1.7 MultiSolidCoverage

MultiSolidCoverage	
Subtype of:	DiscreteCoverage
Definition:	coverage whose domain consists of a collection of solids
Description:	NOTE Solids or their boundaries may be of any shape. Generally, the solids that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage, but this is not required. EXAMPLE Buildings in an urban area could be represented as a set of unconnected GM_Solids each with attributes such as building name, address, floor space and number of occupants.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null

MultiSolidCoverage	
Constraint: domainIsMultiSolid	
Natural language:	domain is a multi solid
OCL:	inv: domainSet.oclIsKindOf(GM_MultiSolid)

6.3.2.1.8 MultiSurfaceCoverage

MultiSurfaceCoverage	
Subtype of:	DiscreteCoverage
Definition:	coverage whose domain consists of a collection of surfaces.
Description:	NOTE In most cases, the surfaces that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage. Surfaces or their boundaries may be of any shape. The boundaries of component surfaces often correspond to natural phenomena and are highly irregular. EXAMPLE A coverage that represents soil types typically has a spatial domain composed of surfaces with irregular boundaries.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: domainIsMultiSurface	
Natural language:	domain is a multi surface
OCL:	inv: domainSet.oclIsKindOf(GM_MultiSurface)

6.3.2.1.9 RectifiedGridCoverage

RectifiedGridCoverage	
Subtype of:	CoverageByDomainAndRange
Definition:	coverage whose domain consists of a rectified grid
Description:	A rectified grid is a grid for which there is an affine transformation between the grid coordinates and the coordinates of a coordinate reference system. NOTE This type can be used for both discrete and continuous coverages.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: domainIsRectifiedGrid	
Natural language:	domain is a rectified grid
OCL:	inv: domainSet.oclIsKindOf(CV_RectifiedGrid)

6.3.2.1.10 ReferenceableGridCoverage

ReferenceableGridCoverage	
Subtype of:	CoverageByDomainAndRange
Definition:	coverage whose domain consists of a referencable grid
Description:	A referencable grid is a grid associated with a transformation that can be used to convert grid coordinate values to values of coordinates referenced to a coordinate reference system. NOTE This type can be used for both discrete and continuous coverages.

ReferenceableGridCoverage	
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: domainIsReferencableGrid	
Natural language:	domain is a referencable grid
OCL:	inv: domainSet.oclIsKindOf(CV_ReferenceableGrid)

6.3.2.2 Data types

6.3.2.2.1 CoverageFunction

CoverageFunction	
Definition:	description how range values at locations in the coverage domain can be obtained
Description:	NOTE The following variants are currently supported: a mapping rule either by inline text or by reference and a grid function that specifies the sequence of the grid points.
Status:	Proposed
Stereotypes:	«union»
URI:	null
Attribute: gridFunction	
Value type:	GridFunction
Definition:	mapping rule for grid geometries
Multiplicity:	1
Attribute: ruleDefinition	
Value type:	CharacterString
Definition:	provides a formal or informal description of the coverage function as text
Multiplicity:	1
Attribute: ruleReference	
Value type:	URI
Definition:	provides a formal or informal description of the coverage function as reference
Multiplicity:	1

6.3.2.2.2 GridFunction

GridFunction	
Definition:	provides an explicit mapping rule for grid geometries
Status:	Proposed
Stereotypes:	«dataType»
URI:	null
Attribute: sequenceRule	
Value type:	CV_SequenceRule
Definition:	description how the grid points are ordered for association to the elements of the values in the range set of the coverage
Multiplicity:	0..1

GridFunction	
Attribute: startPoint	
Value type:	Integer
Definition:	identifies the grid point to be associated with the first record in the range set of the coverage
Description:	If startPoint is omitted it is assumed to be equal to the lowest values in the envelope of the grid geometry.
Multiplicity:	0..*
Collection Constraints:	ordered

6.3.2.3 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.

6.3.2.3.1 Any

Any	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19103 Conceptual Schema Language::ISO 19103:2005 Schema Language::Basic Types::Implementation::Records and Class Metadata

6.3.2.3.2 CV_SequenceRule

CV_SequenceRule	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19123 Coverage geometry and functions::ISO 19123:2005 Coverages::Coverages::Quadrilateral Grid

6.3.2.3.3 CharacterString

CharacterString	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19103 Conceptual Schema Language::ISO 19103:2005 Schema Language::Basic Types::Primitive::Text

6.3.2.3.4 Coverage

Coverage (abstract)	
Package:	INSPIRE Consolidated UML Model::INSPIRE schemas::Generic Conceptual Model::Base Models::Coverages (Base)
Definition:	feature that acts as a function to return values from its range for any direct position within its spatial, temporal or spatiotemporal domain
Description:	EXAMPLE Examples include a raster image, polygon overlay or digital elevation matrix. NOTE In other words, a coverage is a feature that has multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type.

6.3.2.3.5 Integer

Integer	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19103 Conceptual Schema Language::ISO 19103:2005 Schema Language::Basic Types::Primitive::Numerics

6.3.2.3.6 URI

URI	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::Drafts::ISO 19115-1 Metadata - Fundamentals::Citation and responsible party information

6.3.3 Feature catalogue – Coverages (Geometry-value pairs)

Table 7 - Feature catalogue metadata

Feature catalogue name	INSPIRE feature catalogue Coverages (Geometry-Value Pairs)
Scope	Coverages (Geometry-Value Pairs)
Version number	1.0
Version date	2011-06-01
Definition source	INSPIRE data specification Coverages (Geometry-Value Pairs)

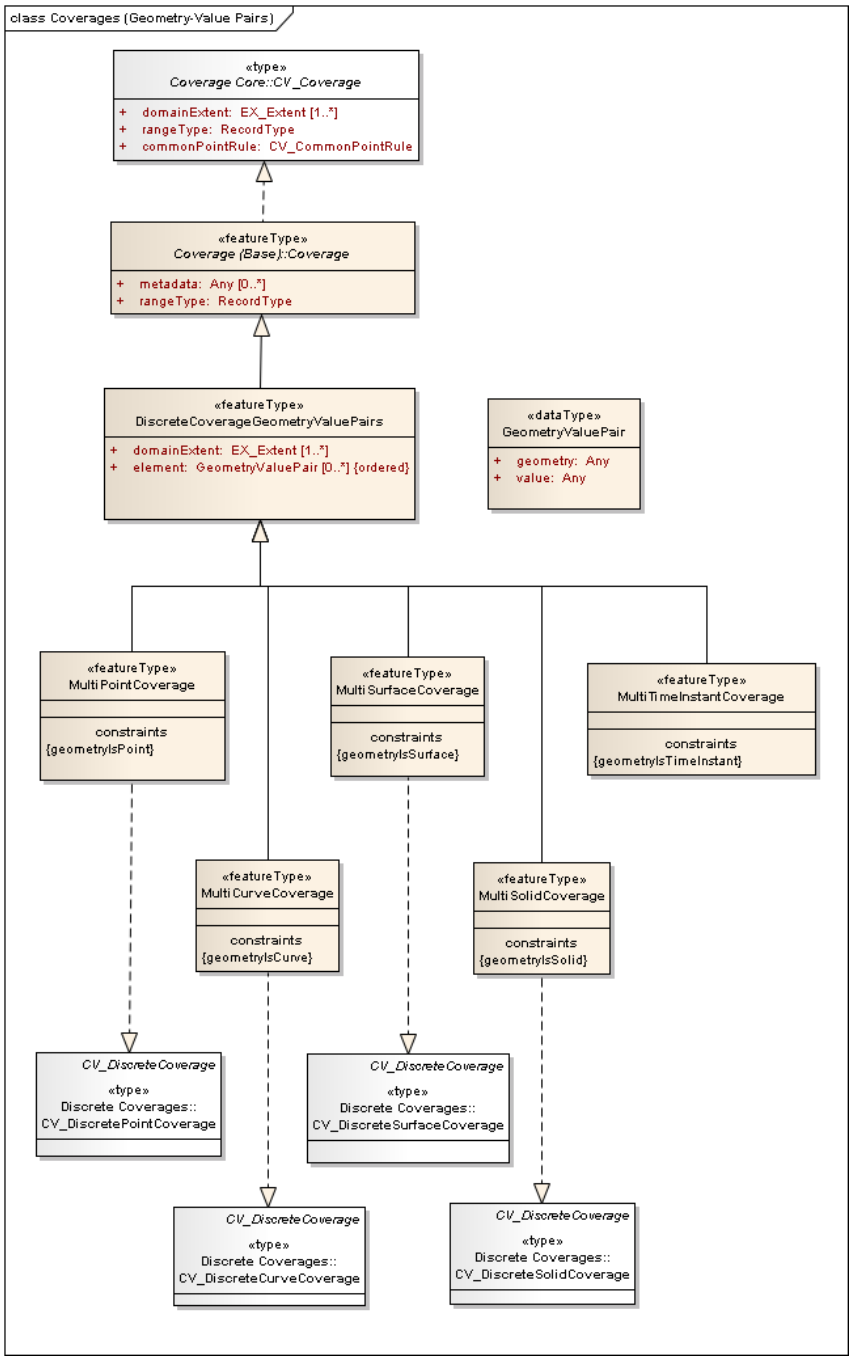

Figure 2 – Coverage representation using geometry/value pairs

Table 8 - Types defined in the feature catalogue

Type	Package	Stereotypes	Section
DiscreteCoverageGeometry-ValuePairs	Coverages (Geometry-Value Pairs)	«featureType»	6.3.2.1.1
GeometryValuePair	Coverages (Geometry-Value Pairs)	«dataType»	6.3.2.2.1
MultiCurveCoverage	Coverages (Geometry-Value Pairs)	«featureType»	6.3.2.1.2
MultiPointCoverage	Coverages (Geometry-Value Pairs)	«featureType»	6.3.2.1.3
MultiSolidCoverage	Coverages (Geometry-Value Pairs)	«featureType»	6.3.2.1.4
MultiSurfaceCoverage	Coverages (Geometry-Value Pairs)	«featureType»	6.3.2.1.5
MultiTimeInstantCoverage	Coverages (Geometry-Value Pairs)	«featureType»	6.3.2.1.6

6.3.3.1 Spatial object types

6.3.3.1.1 DiscreteCoverageGeometryValuePairs

DiscreteCoverageGeometryValuePairs	
Subtype of:	Coverage
Definition:	coverage which provides the geometry value pairs
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Attribute: domainExtent	
Value type:	EX_Extent
Definition:	contains the extent of the domain of the coverage
Description:	NOTE Extents may be specified in space, time or space-time.
Multiplicity:	1..*
Attribute: element	
Value type:	GeometryValuePair
Definition:	list of the geometry/value pairs of the coverage
Multiplicity:	0..*
Collection Constraints:	ordered

6.3.3.1.2 MultiCurveCoverage

MultiCurveCoverage	
Subtype of:	DiscreteCoverageGeometryValuePairs
Definition:	coverage characterized by a finite spatial domain consisting of curves.
Description:	NOTE Often the curves represent features such as roads, railroads or streams. They may be elements of a network. EXAMPLE A coverage that assigns a route number, a name, a pavement width and a pavement material type to each segment of a road system.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: geometryIsCurve	
Natural language:	domain elements are curves

MultiCurveCoverage	
OCL:	inv: element.geometry.oclIsKindOf(GM_Curve)

6.3.3.1.3 MultiPointCoverage

MultiPointCoverage	
Subtype of:	DiscreteCoverageGeometryValuePairs
Definition:	coverage characterized by a finite domain consisting of points
Description:	NOTE 1 This type provides a representation of the point/value pairs. NOTE 2 Generally, the domain is a set of irregularly distributed points. However, the principal use of discrete point coverages is to provide a basis for continuous coverage functions, where the evaluation of the continuous coverage function is accomplished by interpolation between the points of the discrete point coverage. Most interpolation algorithms depend upon a structured pattern of spatial relationships between the points. This requires either that the points in the spatial domain of the discrete point coverage be arranged in a regular way, or that the spatial domain of the continuous coverage be partitioned in a regular way in relation to the points of the discrete point coverage. EXAMPLE A set of hydrographic soundings is a discrete point coverage.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: geometryIsPoint	
Natural language:	domain elements are points
OCL:	inv: element.geometry.oclIsKindOf(GM_Point)

6.3.3.1.4 MultiSolidCoverage

MultiSolidCoverage	
Subtype of:	DiscreteCoverageGeometryValuePairs
Definition:	coverage whose domain consists of a collection of solids
Description:	NOTE Solids or their boundaries may be of any shape. Generally, the solids that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage, but this is not required. EXAMPLE Buildings in an urban area could be represented as a set of unconnected GM_Solids each with attributes such as building name, address, floor space and number of occupants.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: geometryIsSolid	
Natural language:	domain elements are solids
OCL:	inv: element.geometry.oclIsKindOf(GM_Solid)

6.3.3.1.5 MultiSurfaceCoverage

MultiSurfaceCoverage	
Subtype of:	DiscreteCoverageGeometryValuePairs
Definition:	coverage characterized by a finite domain consisting of surfaces

MultiSurfaceCoverage	
Description:	NOTE In most cases, the surfaces that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage. Surfaces or their boundaries may be of any shape. The boundaries of component surfaces often correspond to natural phenomena and are highly irregular. EXAMPLE A coverage that represents soil types typically has a spatial domain composed of surfaces with irregular boundaries.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: geometryIsSurface	
Natural language:	domain elements are surfaces
OCL:	inv: element.geometry.oclIsKindOf(GM_Surface)

6.3.3.1.6 MultiTimeInstantCoverage

MultiTimeInstantCoverage	
Subtype of:	DiscreteCoverageGeometryValuePairs
Definition:	coverage characterized by a finite domain consisting of time instants
Description:	NOTE This type provides a representation of the time instant/value pairs, i.e. represents a time series.
Status:	Proposed
Stereotypes:	«featureType»
URI:	null
Constraint: geometryIsTimeInstant	
Natural language:	domain elements are points
OCL:	inv: element.geometry.oclIsKindOf(TM_Instant)

6.3.3.2 Data types

6.3.3.2.1 GeometryValuePair

GeometryValuePair	
Definition:	describes an element of a set that defines the relationships of a discrete coverage
Description:	Each instance consists of two parts: a domain object from the domain of the coverage to which it belongs and a record of feature attribute values from the range of the coverage to which it belongs.
Status:	Proposed
Stereotypes:	«dataType»
URI:	null
Attribute: geometry	
Value type:	Any
Definition:	the domain object
Multiplicity:	1
Attribute: value	
Value type:	Any

GeometryValuePair	
Definition:	holds the record of feature attribute values associated with the domain object in the property "geometry"
Multiplicity:	1

6.3.3.3 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.

6.3.3.3.1 Any

Any	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19103 Conceptual Schema Language::ISO 19103:2005 Schema Language::Basic Types::Implementation::Records and Class Metadata

6.3.3.3.2 Coverage

Coverage (abstract)	
Package:	INSPIRE Consolidated UML Model::INSPIRE schemas::Generic Conceptual Model::Base Models::Coverages (Base)
Definition:	feature that acts as a function to return values from its range for any direct position within its spatial, temporal or spatiotemporal domain
Description:	EXAMPLE Examples include a raster image, polygon overlay or digital elevation matrix. NOTE In other words, a coverage is a feature that has multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type.

6.3.3.3.3 EX_Extent

EX_Extent	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19115-All Metadata::ISO 19115:2006 Metadata (Corrigendum)::Extent information

6.4 Encoding of Coverages

This sub-clause amends D2.7 version 3.2.

6.4.1 Range encoding

In general, the range of a coverage will not be encoded in XML, but some other binary or text-based format, for example GeoTIFF, netCDF, GRIB, BUFR, JPEG2000, etc.

To comply with the requirements in D2.7 version 3.2, every data specification that uses coverages specifies how range values are encoded and how the information from the conceptual model is represented in the encoding so that a data provider can encode their coverage functions in a way that a receiving system can decode in an unambiguous way.

NOTE D2.7 version 3.2 describes in Annex C additional unresolved issues in this context. These issues are currently still unresolved.

6.4.2 Coverage types

The coverage schemas in ISO 19123 and the GML application schema specify several coverage types for which currently no known Web Coverage Service implementations exist in the market.

EXAMPLE 1 Coverages with a MultiSurface or a MultiCurve as their domain.

In these cases, the option to first derive an implementation model from the conceptual model and then apply the default encoding rule as described in D2.7, version 3.2, B.3.2 is used.

EXAMPLE 2 A MultiSurfaceCoverage may be transformed to a spatial object type with a surface-valued geometry property and a property for the range value in the area of the surface. Other attributes of the coverage may be added to the spatial object, too, or a data set object aggregating the spatial objects representing the coverage may be introduced depending on the thematic requirements.

7 Additional rules for geophysical observations

This sub-clause amends sub-clause 9.4.6 of D2.5 version 3.3. The existing text is kept and the following text is added:

Requirement N17	Whenever the Observation and Measurement standard (O&M in OGC, ISO 19156 in ISO) is used in an INSPIRE application schema, the requirements and recommendations of the document D2.9 "Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development" ("O&M Guidelines") shall be taken into account.

Note that it is planned to maintain the O&M Guidelines as a new part of the INSPIRE modeling framework documents in addition to the current set of documents (D2.5, D2.6 and D2.7).

8 Additional data types used by several themes

This sub-clause amends sub-clause 9.8 of D2.5 version 3.3 with a second application schema for base types.

8.1 Overview

This clause specifies additional types that are used by applications schemas from multiple themes. These will therefore be specified as part of the Generic Conceptual Model.

The new types specified with the Annex II/III data specifications are placed in a separate application schema than the "Base Types" schema to maintain backwards compatibility.

In addition to the types currently included already, there are additional types including:

- Document (Buildings, Land Use)
- DocumentFormatValue (Buildings, Land Use)
- SourceStatusValue (Buildings, Land Use)
- Min/max range types (several themes)
- Party/Organisation/Person types (several themes)
- Status codes (several themes)

Additional candidate types may be identified during the consultation and testing phase.

8.2 Feature Catalogue – Base Types 2

Table 9 - Feature catalogue metadata

Feature catalogue name	INSPIRE feature catalogue Base Types 2

Scope	Base Types 2
Version number	1.0
Version date	2011-06-14
Definition source	INSPIRE data specification Base Types 2

Figure 3 – Legislation references

The *LegislationReference* data type contains a set of properties to cite the legislative instrument and specific articles within the legislation. The following five properties have to be provided as the minimum information for giving a legislation reference:

- *Legal Name:* official name assigned to the legislative instrument
- *Publication Date:* date when the legislative instrument was published in the official journal (NOTE: this is different to when it entered into force)
- *Date of entry into force:* Date when the legislative instrument entered into force
- *Level:* value defining the level at which the legislative instrument is adopted
- *Link to Legislative Instrument:* this is a reference to the online resource

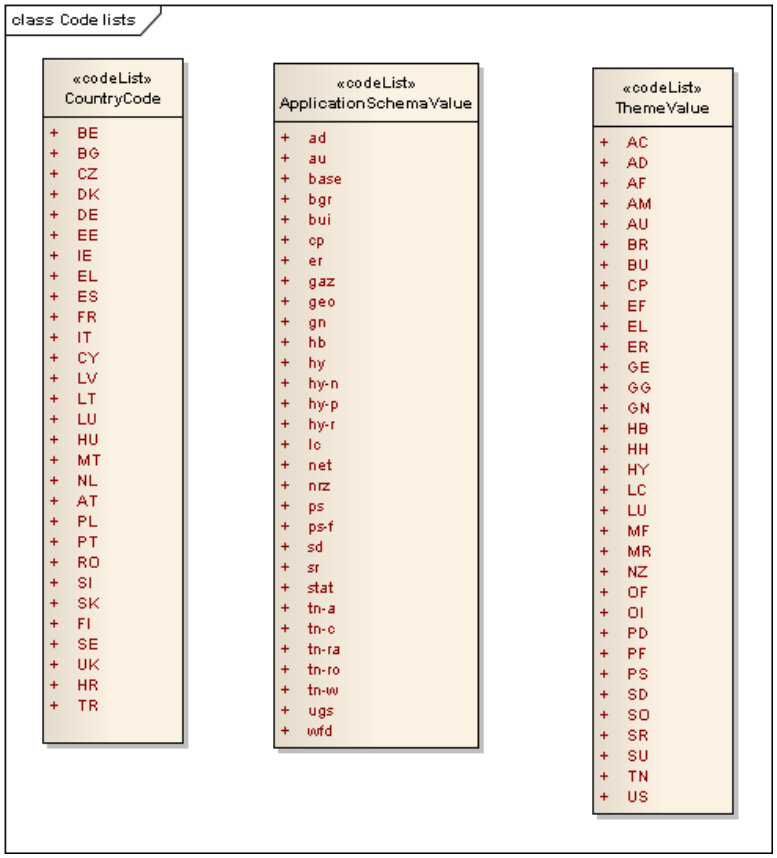

Figure 4 – Additional code lists

Table 10 - Types defined in the feature catalogue

Type	Package	Stereotypes	Section
ApplicationSchemaValue	Base Types 2	«codeList»	8.2.2.1
CountryCode	Base Types 2	«codeList»	8.2.2.2
LegislationLevelValue	Base Types 2	«codeList»	8.2.2.3
LegislationReference	Base Types 2	«dataType»	8.2.1.1
OfficialJournalInformation	Base Types 2	«dataType»	8.2.1.2
ThemeValue	Base Types 2	«codeList»	8.2.2.4

8.2.1 Data types

8.2.1.1 LegislationReference

LegislationReference	
Name:	legislation reference
Definition:	information to unambiguously reference a legal act or a specific part of a legal act
Status:	Proposed
Stereotypes:	«dataType»
URI:	null
Attribute: articleReference	
Name	article reference
Value type:	CharacterString

LegislationReference

Definition:	Reference to article(s) that specify the specific requirement or obligation to establish the object.
Multiplicity:	0..*

Attribute: dateEnteredIntoForce

Name	date entered into force
Value type:	TM_Position
Definition:	Date the legislative instrument entered into force.
Multiplicity:	0..1

Attribute: dateRepealed

Name	date repealed
Value type:	TM_Position
Definition:	Date the legislative instrument was repealed.
Multiplicity:	0..1

Attribute: identificationNumber

Name	identification number
Value type:	CharacterString
Definition:	Code used to identify the legislative instrument
Description:	EXAMPLE 1: 2007/2/EC is the identification number for the INSPIRE Directive EXAMPLE 2: 2008/50/EC is the identification number for the CAFE Directive EXAMPLE 3: 2000/60/EC is the identification number for the Water Framework Directive
Multiplicity:	0..1

Attribute: journalCitation

Name	journal citation
Value type:	OfficialJournalInformation
Definition:	Citation to the Official Journal in which the legislation is published.
Multiplicity:	0..1

Attribute: legalName

Name	legal name
Value type:	CharacterString
Definition:	Official name assigned to the legislative instrument
Description:	EXAMPLE: The official legal name for the INSPIRE Directive is "Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)"
Multiplicity:	1

Attribute: level

Name	name
Value type:	LegislationLevelValue
Definition:	The level at which the legislative instrument is adopted.
Multiplicity:	1

Attribute: linkToLegislativeInstrument

LegislationReference	
Name	link to online version
Value type:	URI
Definition:	Link to an online version of the legislative instrument
Multiplicity:	1
Stereotypes:	«voidable»
Attribute: officialDocumentNumber	
Name	official document number
Value type:	CharacterString
Definition:	Official document number used to uniquely identify the legislative instrument.
Description:	NOTE: An official document number may be assigned to uniquely identify the legislative instrument. EXAMPLE: CELEX Number used to uniquely identify European Union Legislation
Multiplicity:	0..1
Attribute: publicationDate	
Name	publication date
Value type:	TM_Position
Definition:	Date the legislative instrument was published in an official journal.
Multiplicity:	1
Attribute: shortName	
Name	short name
Value type:	CharacterString
Definition:	Short name or alternative title commonly used to identify the legislation
Description:	EXAMPLE 1: INSPIRE Directive is the short name for "Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)" EXAMPLE 2: CAFE Directive is the short name for the Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe" EXAMPLE 3: Water Framework Directive the short name for "Directive 2000/60/EC of the European Parliament and of the Council establishing a framework for the Community action in the field of water policy"
Multiplicity:	0..1
Constraint: If linkToLegislativeInstrument is void journalCitation is mandatory	
Natural language:	If linkToLegislativeInstrument is null journalCitation is mandatory
OCL:	

8.2.1.2 OfficialJournalInformation

OfficialJournalInformation	
Name:	official journal information
Definition:	Full citation to the location of the legislative instrument within the Official Journal.
Status:	Proposed

OfficialJournalInformation	
Stereotypes:	«dataType»
URI:	null
Attribute: ISBN	
Name	ISBN
Value type:	CharacterString
Definition:	International Standard Book Number (ISBN) is an nine-digit number that uniquely identifies the book in which the legislative instrument was published.
Multiplicity:	0..1
Attribute: ISSN	
Name	ISSN
Value type:	CharacterString
Definition:	The International Standard Serial Number (ISSN) is an eight-digit number that identifies the periodical publication in which the legislative instrument was published.
Description:	NOTE: Periodical publilcations are issued in successive parts, usually having numerical or chronological designations and required that each serial publication can be uniquely identified. EXAMPLE: OJ Series in which INSPIRE Directive is published has been assigned the ISSN: 1725-2555
Multiplicity:	0..1
Attribute: linkToJournal	
Name	link to online version
Value type:	URI
Definition:	Link to an online version of the official journal
Multiplicity:	0..1
Attribute: officialJournalIdentification	
Name	official journal identification
Value type:	CharacterString
Definition:	Reference to the location within the official journal within which the legislative instrument was published. This reference shall be comprised of three parts: 1. Journal title 2. Journal volume and/or series number 3. Page number(s)
Description:	EXAMPLE: Official Journal of European Union (OJEU), L108, Volume 50, 1-14
Multiplicity:	1

8.2.2 Code lists

8.2.2.1 ApplicationSchemaValue

ApplicationSchemaValue	
Name:	application schema
Definition:	application schema specified in an INSPIRE data specification

ApplicationSchemaValue	
Description:	SOURCE [INSPIRE Generic Conceptual Model] NOTE This list is not necessarily synchronised with the latest UML model. The current list is based on the model in April 2010 (revision 937), i.e. after alignment with the Implementing Rule from the Annex I process. The code list will be extended to cover Annex II/III, too.
Status:	Proposed
Stereotypes:	«codeList»
Governance:	May not be extended by Member States.
URI:	http://inspire-registry.jrc.ec.europa.eu/codelist/applicationschemavalue
Value: ad	
Name:	Addresses
Value: au	
Name:	AdministrativeUnits
Value: base	
Name:	Base Types
Value: bgr	
Name:	Bio-geographical Regions
Value: bui	
Name:	Buildings
Value: cp	
Name:	CadastralParcels
Value: er	
Name:	Energy Resources
Value: gaz	
Name:	Gazetteer
Value: geo	
Name:	Geology
Value: gn	
Name:	Geographical Names
Value: hb	
Name:	Habitats and Biotopes
Value: hy	
Name:	Hydro - base
Value: hy-n	
Name:	Hydro - Network
Value: hy-p	

ApplicationSchemaValue	
Name:	Hydro - Physical Waters
Value: hy-r	
Name:	Hydro - Reporting
Value: lc	
Name:	Land Cover
Value: net	
Name:	Network
Value: nrz	
Name:	Natural Risk Zones
Value: ps	
Name:	Protected Sites Simple
Value: ps-f	
Name:	Protected Sites Full
Value: sd	
Name:	Species Distribution
Value: sr	
Name:	Sea Regions
Value: stat	
Name:	StatisticalUnits
Value: tn-a	
Name:	Air Transport Network
Value: tn-c	
Name:	Cable Transport Network
Value: tn-ra	
Name:	Railway Transport Network
Value: tn-ro	
Name:	Road Transport Network
Value: tn-w	
Name:	Water Transport Network
Value: ugs	
Name:	Utility and Governmental Services
Value: wfd	
Name:	Water Framework Directive

8.2.2.2 CountryCode

CountryCode	
Name:	country code
Definition:	Country code as defined in the Interinstitutional style guide published by the Publications Office of the European Union.
Status:	Proposed
Stereotypes:	«codeList»
Governance:	May not be extended by Member States.
URI:	http://inspire-registry.jrc.ec.europa.eu/codelist/countrycode
Value: AT	
Name:	Austria
Value: BE	
Name:	Belgium
Value: BG	
Name:	Bulgaria
Value: CY	
Name:	Cyprus
Value: CZ	
Name:	Czech Republic
Value: DE	
Name:	Germany
Value: DK	
Name:	Denmark
Value: EE	
Name:	Estonia
Value: EL	
Name:	Greece
Value: ES	
Name:	Spain
Value: FI	
Name:	Finland
Value: FR	
Name:	France
Value: HR	
Name:	Croatia
Value: HU	
Name:	Hungary

CountryCode	
Value: IE	
Name:	Ireland
Value: IT	
Name:	Italy
Value: LT	
Name:	Lithuania
Value: LU	
Name:	Luxembourg
Value: LV	
Name:	Latvia
Value: MT	
Name:	Malta
Value: NL	
Name:	Netherlands
Value: PL	
Name:	Poland
Value: PT	
Name:	Portugal
Value: RO	
Name:	Romania
Value: SE	
Name:	Sweden
Value: SI	
Name:	Slovenia
Value: SK	
Name:	Slovakia
Value: TR	
Name:	Turkey
Value: UK	
Name:	United Kingdom

8.2.2.3 LegislationLevelValue

LegislationLevelValue	
Name:	legislation level
Definition:	The level at which a legal act or convention has been adopted.

LegislationLevelValue	
Status:	Proposed
Stereotypes:	«codeList»
Governance:	May be extended by Member States.
URI:	http://inspire-registry.jrc.ec.europa.eu/codelist/legislationlevelvalue
Value: european	
Definition:	A legal act of the European Union.
Value: international	
Definition:	An international legal act or convention.
Value: national	
Definition:	A legal act at national level.
Value: sub-national	
Definition:	A legal act at sub-national level.
Description:	NOTE The sub-national level encompasses both regional or local legislative instruments.

8.2.2.4 ThemeValue

ThemeValue	
Name:	spatial data theme
Definition:	grouping of spatial data according to Annex I, II and III of the INSPIRE Directive
Description:	SOURCE [INSPIRE Generic Conceptual Model]
Status:	Proposed
Stereotypes:	«codeList»
Governance:	May not be extended by Member States.
URI:	http://inspire-registry.jrc.ec.europa.eu/codelist/themevalue
Value: AC	
Name:	Atmospheric conditions
Value: AD	
Name:	Addresses
Value: AF	
Name:	Agricultural and aquacultural facilities
Value: AM	
Name:	Area management/restriction/regulation zones and reporting units
Value: AU	
Name:	Administrative units
Value: BR	
Name:	Bio-geographical regions
Value: BU	
Name:	Buildings

ThemeValue	
Value: CP	
Name:	Cadastral parcels
Value: EF	
Name:	Environmental monitoring facilities
Value: EL	
Name:	Elevation
Value: ER	
Name:	Energy resources
Value: GE	
Name:	Geology
Value: GG	
Name:	Geographical grid systems
Value: GN	
Name:	Geographical names
Value: HB	
Name:	Habitats and biotopes
Value: HH	
Name:	Human health and safety
Value: HY	
Name:	Hydrography
Value: LC	
Name:	Land cover
Value: LU	
Name:	Land use
Value: MF	
Name:	Meteorological geographical features
Value: MR	
Name:	Mineral resources
Value: NZ	
Name:	Natural risk zones
Value: OF	
Name:	Oceanographic geographical features
Value: OI	
Name:	Orthoimagery

ThemeValue	
Value: PD	
Name:	Population distribution, demography
Value: PF	
Name:	Production and industrial facilities
Value: PS	
Name:	Protected sites
Value: SD	
Name:	Species distribution
Value: SO	
Name:	Soil
Value: SR	
Name:	Sea regions
Value: SU	
Name:	Statistical units
Value: TN	
Name:	Transport networks
Value: US	
Name:	Utility and governmental services

8.2.3 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.

8.2.3.1.1 CharacterString

CharacterString	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19103 Conceptual Schema Language::ISO 19103:2005 Schema Language::Basic Types::Primitive::Text

8.2.3.1.2 TM_Position

TM_Position	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19108 Temporal schema::ISO 19108:2006 Temporal Schema::Temporal Reference System

8.2.3.1.3 URI

URI	
Package:	INSPIRE Consolidated UML Model::Foundation Schemas::ISO TC211::ISO 19136 GML::basicTypes

9 Feature associations and identifiers

The INSPIRE application schemas include navigable association roles that have target types which do not contain an Identifier-valued property. In addition, not every feature instance will have such an identifier.

Currently, D2.7 version 3.2 contains the recommendation to "include the namespace and the local identifier part of the" URI that is used to identify - and reference - the object. This needs to be amended to cover all cases.

The following recommendation replaces recommendation 17 of D2.7 version 3.2.

Recommendation N10 To reference a spatial object or a specific version of a spatial object its persistent URI based on its INSPIRE identifier (see Recommendation 15) should be used. For spatial objects without an INSPIRE identifier a URI based on the gml:id of the GML representation of the spatial object should be used.

10 Metadata

The following table replaces Table 10 in Clause 18 of D2.5 version 3.3.

Table 1 – Mandatory and conditional common metadata elements

Metadata element	Multiplicity	Condition
Coordinate Reference System	1	
Temporal Reference System	0..*	Mandatory, if the spatial data set or one of its feature types contains temporal information that does not refer to the Gregorian Calendar or the Coordinated Universal Time.
Encoding	1..*	
Character Encoding	0..*	Mandatory, if an encoding is used that is not based on UTF-8.
Error! Reference source not found.	0..*	Mandatory, if the data set includes types from the Generic Network Model and does not assure centreline topology (connectivity of centrelines) for the network.

11 Spatial Data Set

The amends the notes for spatial object type SpatialDataSet in the application schema BaseTypes, see D2.5 version 3.3, 9.8.2.1

Definition	Identifiable collection of spatial data.
Description	NOTE The type SpatialDataSet is offered as a pre-defined type for spatial data sets. The scope of this type is packaging pre-defined data sets for the non-direct access download service. It may be reused in thematic application schemas, but if it does not fit an application-schema-specific data set object should be modelled. This type may have the same name and definition like this type, but may have a different set of properties. This type specifies three properties: an external object identifier, a container for metadata (may be void), and an association to zero or more spatial objects.

Annex A : Code-list-related requirements and recommendations currently included in the IR-ISDSS, D2.5 and D2.7

This annex summarises the requirements and recommendations currently contained in the Implementing Rules for interoperability of spatial data sets and services (IR-ISDSS), the Generic Conceptual Model (D2.5),[2] and the Encoding Guidelines (D2.7)[3].

A.1 Requirements on code lists in IR-ISDSS

There has been an extended discussion between the TWGs, DT DS, CT and the Member States during the Annex I development process on how code list requirements should be included in the IR-ISDSS.

In the end, due to an intervention from the EC Legal Service during the inter-service consultation, the preferred option of making a reference to the INSPIRE code list register, could not be used. Instead, the IR on interoperability of spatial data sets and services (IR-ISDSS) now explicitly includes all allowed values for each code lists. Thus, the IR factually becomes the common INSPIRE code list register.

The main argument for this intervention was that, if the Commission puts a requirement on the Member States to use only certain values (see Article 6(3) below), it must be guaranteed that no change to the allowed values will be possible "outside" of the legal procedure (Comitology with scrutiny) specified for the IR-ISDSS. In other words, changes to the code lists can only be made through formal amendments of the legal act.

The Implementing Rules on interoperability of spatial data sets and services (IR-ISDSS) and its amendment regarding code lists includes the following requirements on code lists:

Article 2

Definitions

(…)

5. 'code list' means an open enumeration that can be extended,
6. 'data type' means a descriptor of a set of values that lack identity, in accordance with ISO 19103,
7. 'enumeration' means a data type whose instances form a fixed list of named literal values. Attributes of an enumerated type may only take values from this list,

(…)

(…)

Article 4

Types for the Exchange and Classification of Spatial Objects

(…)

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 language-neutral mnemonic codes for computers.

(…)

[2] http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/D2.5_v3_3.pdf
[3] http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/D2.7_v3.2.pdf

Article 6

Code Lists and Enumerations

1. Code lists shall be of one of the following types, as specified in Annex II:
   (a) code lists that shall not be extended by Member States;
   (b) code lists that may be extended by Member States.
2. Where a Member State extends a code list, the allowed values of the extended code lists shall be made available in a register.
3. Attributes or association roles of spatial object types or data types whose type is a code list as defined in Article 6 (1)(a) may only take values from the lists specified for the code list.

   Attributes or association roles of spatial object types or data types whose type is a code list as defined in Article 6 (1)(b) may only take values that are valid according to the register in which the code list is managed.
4. 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.

A typical specification of a code list shown below.

Condition Of Facility (ConditionOfFacilityValue)

The status of a facility with regards to its completion and use.

This code list shall be managed in a common code list register.

Allowed values for the code list ConditionOfFacilityValue

Value	Definition
disused	The facility is not used.
functional	The facility is functional.
projected	The facility is being designed. Construction has not yet started.
underConstruction	The facility is under construction and not yet functional. This applies only to the initial construction of the facility and not to maintenance work.

A.2 Requirements and recommendations from the GCM (D2.5)

The Generic Conceptual Model v3.3 includes the following recommendation and requirements on code lists:

Recommendation 4	In the case of an attribute type with coded values, an enumeration or code list should be used. If the set of allowed values may be extended by user communities or without a major revision of the data specification, a code list should be used. If the set of allowed values is fixed, an enumeration should be used. For code lists, the use of a code list managed in the INSPIRE code list register should be mandated.

(…)

Requirement 24	Values of code lists and enumerations shall have no initial value and the “attribute name” shall conform to the rules for attributes names, i.e. be without whitespace and where only the first letter of each word after the

	first word that is combined in the value name is capitalised. Exceptions are words that consist of all uppercase letters (acronyms).

Requirement 25	To distinguish the different types of code lists, a tagged value “codeList” shall be provided for each code list that is intended to be centrally managed in the INSPIRE code list register. The value shall be a persistent URI identifying the code list. If the value is set, then any conformant data shall use only values provided by the INSPIRE code list register.

The value of the tagged value “codeList” will be preliminarily set to "urn:x-inspire:def:codeList:INSPIRE:<name of the class>". This value may change once the INSPIRE code list register has been established.

A.3 Naming conventions

In Annex I, it was agreed that the class name for enumerations and code lists should include the suffix “Value”, e.g. IndustryTypeValue or SoilTypeValue (see discussion at http://inspire-twg.jrc.ec.europa.eu/jira/browse/DS-195).

A.4 Recommendations and requirements from Encoding Guidelines (D2.7)

The Encoding Guidelines v3.2 include the following recommendation and requirements on code lists:

B.2 Use of URIs

Within INSPIRE specifications, metadata and spatial data require access to resources, such as schemas, documents, reference systems, spatial objects, definitions, code lists, spatial data services, etc. In order for the larger community of users of spatial data and developers of software components, a persistent mechanism to uniquely reference resource – independent of their physical location in the network.

Uniform Resource Identifiers (URIs) as defined in IETF RFC 3986 are intended to serve as these persistent, location-independent resource identifiers.

Recommendation 8	Identifiers of resources should be URIs in the “http” scheme.

Recommendation 9	While it is not strictly a requirement that the resource can be accessed via the HTTP protocol using its http URI, it is strongly recommended that this is the case.

The use of HTTP includes support for content negotiation to access a resource in different languages and/or different encodings, if supported.

(…)

Requirement 10	All INSPIRE code lists shall be assigned a tagged value "asDictionary" with the value "true". Instance data shall reference a GML dictionary that encodes the valid value in the code list.

NOTE 2 As a result, the code list values are managed only in registries outside of the application schema. All instances have to reference the registry to allow applications to evaluate the code list value and its validity, its title in the official languages or its definition.

NOTE 3 If a code list relevant for an INSPIRE data specification is already maintained by an international organisation, then INSPIRE will simply adopt that code list and reference it. However, if the international organisation does not provide a representation of the code lists as a GML dictionary it may be required to maintain such a representation in an INSPIRE registry.

(…)

B.4.4 Encoding of a reference to a registered item

Recommendation 18 URIs should be used to encode item identifiers of items in ISO 19135-compliant registers and to reference such items.

The URIs should use the following structure:

http://inspire-registry.jrc.ec.europa.eu/<register>/<item class>/<item identifier>

where
- <item class> is the name of the item class (ISO 19135: RE_ItemClass) of the registered item;
- <register> is the name of the register (ISO 19135: RE_Register);
- <item identifier> is the item identifier of the registered item (ISO 19135: RE_RegisterItem).

Colons should not be used in <item class>, <register> or <item identifier> values.

See also Issue 1, for now this is a preliminary URI pattern and implemented at this time.

Other URIs may be used, too, for items managed in external registers. For registers with stable URIs as identifiers, these URIs may be used instead. For external registers that do not provide stable URIs, the “inspire”-URIs will be used, but the <register> will identify the external register and authority.

In principle, INSPIRE will adopt existing registers and their items whenever possible as long as they are applicable to INSPIRE, managed in a structured way by a competent international organisation, and are accessible from the INSPIRE SDI. In this context, several aspects will have to be considered/addressed:

- Organisational: The adoption of a third-party register creates a dependency and requires an agreement between INSPIRE and the other organisation. This includes intellectual property rights aspects, notification of changes, etc.
- Operational: To address the interoperability requirements there must be ways how users of INSPIRE data will be able to understand items managed in the external registers. This may require setting up registries for such registers to make these items available within INSPIRE. This will require further work.
- Technical: To allow referencing items in external registers, the URI scheme used in D2.7 will be adapted to allow this.

EXAMPLE 1 http://inspire-registry.jrc.ec.europa.eu/IFCD/featureConcept/125 would be a reference to a feature concept in the INSPIRE Feature Concept Dictionary. The item identifier 125 is unique and identifies the item within the register. It should be noted that the name of the feature concept cannot be used as the item identifier as over time the same name may be associated with different concepts; e.g., when the definition of a feature concept is amended.

EXAMPLE 2 OGC provides a URN namespace to reference coordinate reference systems, e.g., urn:ogc:def:crs:EPSG::4258 would be a reference to the geographic coordinate reference system ETRS89 in the EPSG coordinate reference system register.

This applies to all items that will be managed in registers: terms, feature types, coordinate reference systems, coordinate operations, units of measurements, identifier namespaces, application schemas, etc. The list of item classes will be compiled during the data specification process.

NOTE 1 The intention of this recommendation is to harmonise the encoding style across the different data specifications where the GML encoding rule leaves freedom.

NOTE 2 The use of other URIs is intended to support the usage of URIs that are already commonly used and supported by software components or specified in other standards.

A.5 Enumerations vs. code lists (from GCM)

In section 9.4.4 of the Generic Conceptual Model v3.3, the differences between code lists and enumerations are described as follows (our emphasis):

There are at least two cases **where code lists may be more suitable**:

- the list of possible values of an attribute are difficult to harmonise and some data providers may have to use sub-sets or extensions of the harmonised list
- the list of possible values is likely to evolve, some other values may have to be added later, either because of new user requirements or because of upgrading of existing data.

To reflect the different characteristics of code lists **two types of code list governance** are distinguished:

- code lists that are managed centrally in the INSPIRE code list register and only values from that register may be used, and
- code lists that may be extended by data providers as long as data providers maintain their extensions in a national code list register and publish its content.

NOTE The details of the publication of national code list registers has not been specified yet and depends on the implementation of the INSPIRE code list register.

The following **examples** highlight this difference

EXAMPLE 2 For the origin of a hydrographic feature, two values are distinguished:
- man-made
- natural

As this list of possible values may be considered as exhaustive, it is not likely to change; i.e., it may be specified as an **enumeration** in the application schema.

EXAMPLE 3 To describe the hydrological persistence of a body of water, the following values are currently in use:
- dry
- ephemeral
- intermittent
- perennial

As this list of possible values may not be exhaustive, the user requirements may evolve and this list may have to be enriched, it is better to specify it as a code list with the above values as initial values of the **code list**. Since the allowed values should be the same for all applications, this code list would be **centrally managed**.

EXAMPLE 4 In contrast to EXAMPLE 3, the **code list** containing the allowed values for designation schemes for protected sites (currently containing international designation schemes such as Natura 2000 or Ramsar) should be **extensible** so that also national designation schemes can be used.