Agency for Data Supply and Efficiency

2016.1 Fitness for purpose Annex III

Danish comments

Author	Agency for Data Supply and Efficiency (SDFE)
Creation date	08-06-2017

Table of Contents

1 Scope

This document summarizes and gives additional and clarifying remarks on the presentation given by the Danish Agency of Data Supply and Efficiency (SDFE) to the Reflection Group ([1]).

Additionally, this document contains the comments from the Danish Agency of Data Supply and Efficiency, the Danish National Point of Contact for INSPIRE, on the report on INSPIRE Fitness for purpose created on behalf of the German National Point of Contact for INSPIRE ([2]).

2 Executive summary

Denmark is of the opinion that the INSPIRE guidance documents should be updated to clearly state that an encoding rule, including the INSPIRE default encoding rule, can at most be recommended, never mandatory. Denmark stands by the flattening approach presented earlier to the Reflection Group and additionally proposes the use of "conversion rules for decoding (voidable) properties" as a way to simplify encodings.

Denmark mostly agrees with the comments formulated by the German National Point of Contact for INSPIRE, including the proposed change to the implementing rules, REC-12 (Simply View Service Layer Structure). However, Denmark is of the opinion that the encoding of measures should not be changed.

The Danish recommendations should be viewed in the light of an overall need for ensuring a balance between harvesting the value of interoperability, implementing in a cost efficient way and providing user relevant solutions.

3 "The Danish suggestion" summarized and clarified

In [1], SDFE showed an example of how to create a simpler GML application schema from the INSPIRE theme Area management / restriction / regulation zones & reporting units. The different steps in this exercise are depicted in Figure 1 and can be configured and automatically executed[1].

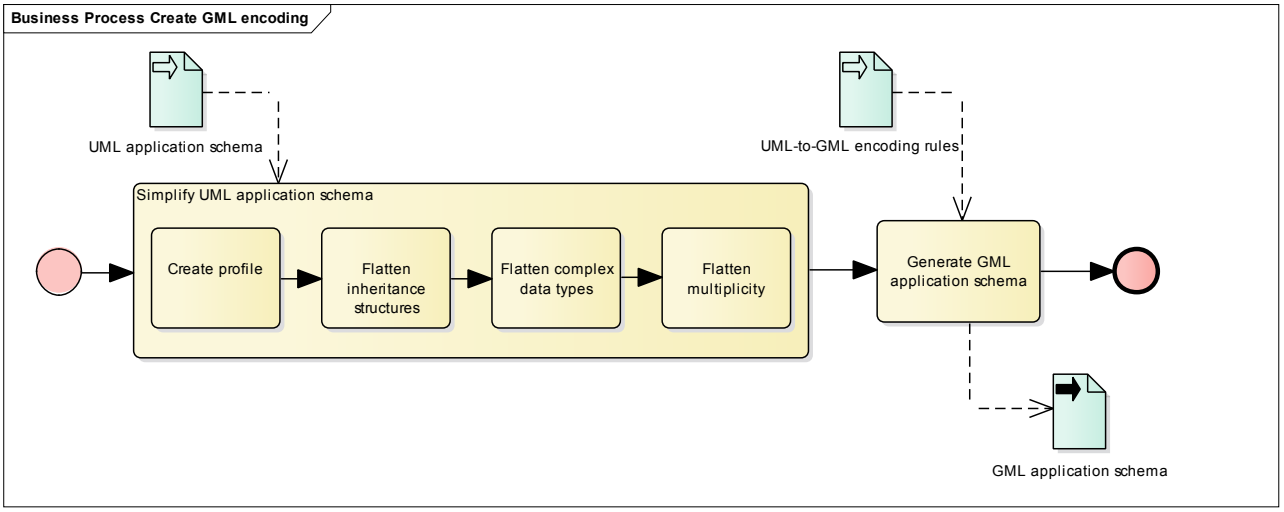


Figure 1: Overview of the process of creating an encoding.

More information about these steps is given in the next sections. For a concrete example, see [1].

[1] See http://shapechange.net/transformations/profiler/, http://shapechange.net/transformations/flattener/#rule-trf-cls-flatten-inheritance, http://shapechange.net/transformations/flattener/#rule-trf-prop-flatten-types and http://shapechange.net/transformations/flattener/#rule-trf-prop-flatten-multiplicity.

3.1 Profiling

Creating a profile[2] of the INSPIRE data model is done on the basis of **knowledge of the source data set and its potential use**. For each spatial object type and each of its properties, it is decided whether or not it is a part of the profile.

3.1.1 Property not maintained

When it is known that a certain property is not contained in the source data set, then a decision could be made not to include this property in the profile. Instead, this knowledge could be provided by other means, such as by mentioning it in the metadata for the data set. In fact, this knowledge is a conversion rule for decoding, see also section 3.5.2.

When decoded into the full INSPIRE data model, all the instances of this property would be void and would receive void reason value "Unpopulated"[3] in the INSPIRE default encoding, which is described in [4].

In this case, it is assumed that the property actually is voidable. However, the majority of properties in the INSPIRE data model are voidable, as one recommendation for the creation of INSPIRE data specifications states that *all properties of spatial object types except those without which a spatial object is not meaningful should be voidable* ([5, p. 47]).

3.1.2 Property is constant

When it is known that a certain property has the same value for all spatial objects in a data set, the same applies: the property could be excluded from the profile and this information should be provided as a conversion rule for decoding, see also section 3.5.2.

3.1.3 Property is never present

When it is known that an optional property never occurs in the data set, the same applies: the property could be excluded from the profile and this information should be provided as a conversion rule for decoding, see also section 3.5.2.

3.1.4 Property of no relevance

When it is known that a certain property has no relevance for a given use case, a decision could be made not to include this property in the profile designed for this use case.

When decoded into the full INSPIRE data model (see also section 3.5.2), all the instances of this property would be void. At first sight, the appropriate void reason value would be "Withheld". However, the semantics from this value have been changed in the INSPIRE code list in comparison to the GML 3.2 specification. In the GML 3.2 specification, the definition is simply "*the value is not divulged*" ([6, p. 37]). The definition in INSPIRE is "*The characteristic may exist, but is confidential and not divulged by the data provider*". ([5, p. 66]).

Denmark proposes that the semantics of "Withheld" are aligned with the semantics in the GML specification. A possibility may be to make the code list hierarchical (see also [5, p. 137]) and add a value "Confidential" as a sub-value of "Withheld". Denmark would at the same time also like to encourage the MIG to provide guidance on how exactly to encode void reason values, see [7].

3.2 Flattening inheritance structures

This step leads to what is described in *REC-05: Reduce use of substitution groups* ([2]).

---

[2] According to ISO 19106, a profile is *a set of one or more base standards or subsets of base standards, and, where applicable, the identification of chosen clauses, classes, options and parameters of those base standards, that are necessary for accomplishing a particular function* ([3]). Thus, a profile of the INSPIRE data model is a subset of the INSPIRE data model.

[3] From http://inspire.ec.europa.eu/codelist/VoidReasonValue/Unpopulated: *The characteristic is not part of the dataset maintained by the data provider. However, the characteristic may exist in the real world. NOTE The characteristic receives this value for all objects in the spatial data set.*

3.3 Flattening complex data types

This step leads to what is described in *REC-01: Flattening of complex structures* and in *REC-04: Simplify xs:choice elements in data models* ([2]).

3.4 Flattening multiplicity

Flattening the multiplicity is done on the basis of **knowledge of the source data set**. E.g. if it is known, by analysing the data set, that a feature in practice has no more than 3 names, then the multiplicity of the property name can be flattened to 3. Or to 4 or 5, to be on the safe side, as data sets are subject to change.

This number of maximum occurrence can be configured on a property by property basis, and a default maximum occurrence can be set for a certain data set, or collection of data sets.

3.5 Encoding & decoding

3.5.1 Encoding

The INSPIRE guidance document “D2.7: Guidelines for the encoding of spatial data” *specifies requirements and recommendations for the encoding of spatial data for the purpose of data interchange of systems in INSPIRE* ([4, p. 6]). This document was developed to be *a basis for the development of the thematic data specifications that will serve as technical basis for the legal text of the INSPIRE Implementing Rules* ([4, p. 2]).

As an INSPIRE guidance document, [4] *does not represent an official position of the European Commission, and as such cannot be invoked in the context of legal procedures* ([4, p. 2]). This means that the requirements set out in [4] are not legally binding requirements unless they are repeated in the implementing rules. More specific, this means that requirement 2 - “Every data specification shall specify a mandatory encoding rule that has to be supported for the spatial data of that theme” (see also Figure 2) - is not legally binding. This also means that the flattening approach as described above is perfectly legitimate, as also pointed out in [8, p. 21].

Denmark proposes that guidance document “D2.7: Guidelines for the encoding of spatial data” is updated. Requirement 2 should be changed to “Every data specification shall specify a recommended encoding rule for the spatial data of that theme” and every occurrence of “mandatory encoding rule” should be changed to “recommended encoding rule”. On the same occasion, the style of the technical requirements should be updated so it follows the guidelines set out in [9, p. 3].

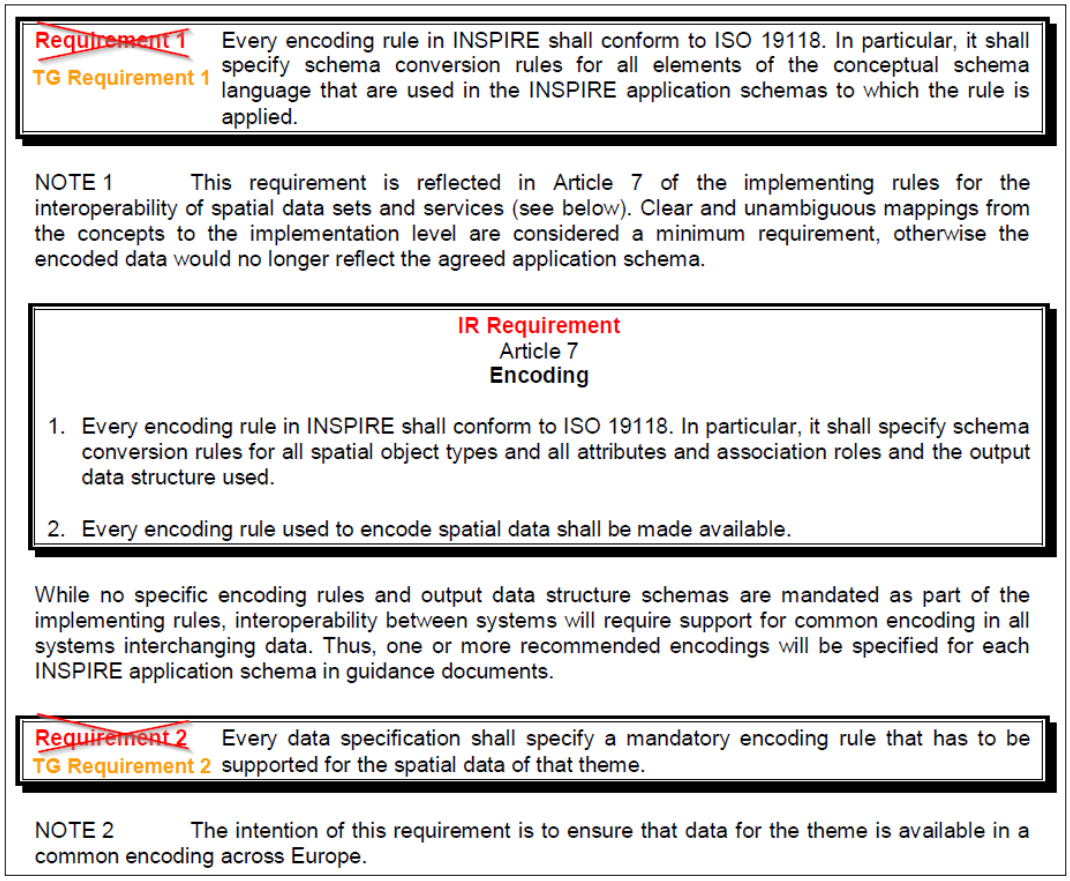


Figure 2: Implementing Rules Requirement regarding encoding and the first two requirements regarding encoding defined by [4, p. 15]. The styles for the TG requirements and the recommendations should be updated in [4] so it follows the guidelines set out in [9, p. 3].

3.5.2 Decoding

ISO 19118:2011 states the following ([10, p. 15]):

An encoding rule shall in general specify the following:

[…]

d) conversion rules, called the mapping, for converting data in the instance model to the exchange format:

1) conversion rules for encoding

2) if necessary, conversion rules for decoding

Denmark is of the opinion that an encoding rule that is based on a profile of an INSPIRE UML application schema and that provides conversion rules for how to decode the properties excluded from that profile (as described in 3.1 Profiling) conforms to the INSPIRE directive and implementing rules.

These conversion rules can also be used as specification for the transformation of INSPIRE data in an alternative encoding to data in the INSPIRE default encoding.

Denmark proposes that INSPIRE guidance document D2.7 is updated to include the principles of decoding properties excluded from a profile and encourages the MIG to come up with a consistent way of documenting such conversion rules.

4 Comments on the German report

[2] contains the recommendations for changes to INSPIRE listed in the table below.

Category	Recommendation
IR-Rec[4]	REC-12: Simply View Service Layer Structure
TG-Rec[5]	REC-09: Provide code list references in XML Application Schemas
	REC-11: Split features with large (data set) scope
Alt-Rec[6]	REC-01: Flattening of complex structures
	REC-02: Simplify attributes representing measures
	REC-03: Remove obligation to provide UnitOfMeasure for attributes that are Voidable
	REC-04: Simplify xs:choice elements in data models
	REC-05: Reduce use of substitution groups
	REC-06: Use alternative encoding for specific elements
	REC-07: Avoid features with multiple geometry attributes
	REC-10: Provide alternative data models for View Services
	REC-13: Provide more encoding examples
Arch-Rec[7]	REC-08: Make available transformation infrastructure to support alternative encoding

4.1 Section 4.1, the use case of data management

First of all, it should be noted that the implementation of INSPIRE does not mean that data providers should maintain their data structured according to the INSPIRE data models. So it is not relevant to investigate the impact of any recommendations on data management.

An additional comment is that this section does not take into account the possibility that data providers may manage their spatial data with mainstream relational database management systems (RDBMS) enhanced with an extra component to support spatial data. Examples are Oracle with the Oracle Spatial and Graph features and PostgreSQL with the PostGIS extension.

In RDBMSs, maintaining relations between tables is common practice. In RDBMSs with a spatial component, maintaining spatial object tables with multiple geometry columns is not a problem. RDBMSs do not natively support inheritance but well-known mapping patterns exists ([11]) so that data can be managed using an RDBMS and can be interchanged according to an application schema that uses inheritance.

4.2 REC-01: Flattening of complex structures

According to [2, p. 15], a negative consequence of flattening of complex structures would be the following:

The thematic structure of the data is not any more reflected in the encoding structure, so interpreting a dataset that has been flattened from a very complex structure (e.g. multiple levels) could be more difficult.

The Agency for Data Supply and Efficiency has experience with providing an unambiguous mapping - that thus can be processed without the need of human intervention – between

- a UML model and an application schema in XML[8]
- a UML model and an application schema in JSON.

This is done by adding references in the application schema itself to the corresponding UML model constructs. Every model construct has an id that is globally unique, and therefore it is possible to make this

[4] Changes to Implementing Rule
[5] Changes to Technical Guidance
[6] Extension of Technical Guidance
[7] Extension to INSPIRE architecture
[8] Not a GML application schema, but in SDFE called a ”replication schema”, which an application schema in XML that does not use GML.

unambiguous mapping. This method is described in [12], works also when complex structures are flattened and is implemented in ShapeChange in a way that would make it straightforward to implement it in GML application schemas. An example is given below.

This means that the disadvantage described above could be eliminated by providing this explicit and unambiguous mapping in the application schema. On the other hand, if the flattening is done in a stringent way, it is clear from the names in the GML application schema how to interpret a dataset.

```
  <element name="MySpatialObject" type="sn:MySpatialObjectType">
    <annotation>
      <appinfo>EAID_427F2895_4168_4ce3_A8CB_015DAF1AD996</appinfo>
    </annotation>
  </element>
  <complexType name="MySpatialObjectType">
    <sequence>
      <element minOccurs="0" name="myFirstProperty" type="string">
        <annotation>
          <appinfo>EAID_A690FAE1_99C6_4eea_B98F_D7390F47C0BB</appinfo>
        </annotation>
      </element>
      <element minOccurs="0" name="mySecondProperty" type="string">
        <annotation>
          <appinfo>EAID_03AF1E12_0105_48c3_B5C2_2A7098761C1A</appinfo>
        </annotation>
      </element>
      <element minOccurs="0" name="myComplexProperty_attribute1" type="string">
        <annotation>
<appinfo>EAID_97C91B31_181D_46c0_93BD_F90861930993.EAID_6CEB0356_5CBF_4159_B96B_A2489DD2DAC8</app
info>
        </annotation>
      </element>
      <element minOccurs="0" name=" myComplexProperty_attribute2" type="string">
        <annotation>
<appinfo>EAID_97C91B31_181D_46c0_93BD_F90861930993.EAID_9AB90AE0_9F85_4164_9B25_8EB2139D65A5</app
info>
        </annotation>
      </element>
    </sequence>
  </complexType>
```

4.3 REC-02: Simplify attributes representing measures

In the opinion of Denmark, the encoding of measures should not be changed. Instead, GIS clients should be improved by the vendors or by the community so they can understand not only the measured values but also the associated unit of measures.

XML attributes are together with XML elements the basic building blocks of XML. They are in themselves not to be considered to be complex.

In the simplest compliance level of the GML simple features profile standard ([13]), SF-0, the GML measure type is included in the list of allowed types for non-spatial property: *Non-spatial properties are limited to being of type: integer, measurement, date, boolean, binary, URI, character or real* ([13, p. 2]), see also Table 1. Furthermore, this standard *strongly recommends that real-valued properties be encoded using the gml:MeasureType* ([13, p. 27]).

A third argument against REC-02 is pointed out already in [2, p. 16]: *often tools will still be able to work with the value even if they don't recognise the unit of measurement (given that it is the same throughout a data set).*

Table 1: Compliance level capabilities matrix ([13, p. 3]).

	Level SF-0	Level SF-1	Level SF-2
restricted set of built-in non-spatial property types	Yes[1]	Yes[1]	No
restricted set of spatial property types	Yes[2]	Yes[2]	Yes[2]
user-defined property types	No	Yes	Yes
use of nillable and xsi:nil	No	Yes	Yes
cardinality of properties	0 or 1	0…unbounded	0…unbounded
non-spatial property values references	Yes[3]	Yes[3]	Yes
spatial property values references	Yes[3]	Yes[3]	Yes

1. string, integer, measurement, date, real, binary, boolean, URI
2. Point, Curve (LineString), Surface (Polygon), Geometry, MultiPoint, MultiCurve, MultiSurface, MultiGeometry
3. In levels 0 and 1, remote values for properties are supported **only** through the use of the type gml:ReferenceType. The more generalized GML property-type pattern allowing mixed inline and by-reference encoded property values within the same instance document is disallowed.

4.4 REC-03: Remove obligation to provide UnitOfMeasure for attributes that are Voidable

Attribute uom, containing the unit of measure, is indeed required. The fact that the element may contain an attribute `xsi:nil="true"` does not change this, according to the XML Schema Recommendation ([14]).

`gml:MeasureType` is defined as follows:

```
<complexType name="MeasureType">
  <simpleContent>
    <extension base="double">
      <attribute name="uom" type="gml:UomIdentifier" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

As explained in the previous section, in the opinion of Denmark, the existing encoding for measures should not be changed.

Denmark proposes that the suggestion given in [13, p. 17] is followed: *In an instance document, if the unit of measure is not known, then the value of the uom attribute may be set to* **unknown** *to indicate this*.

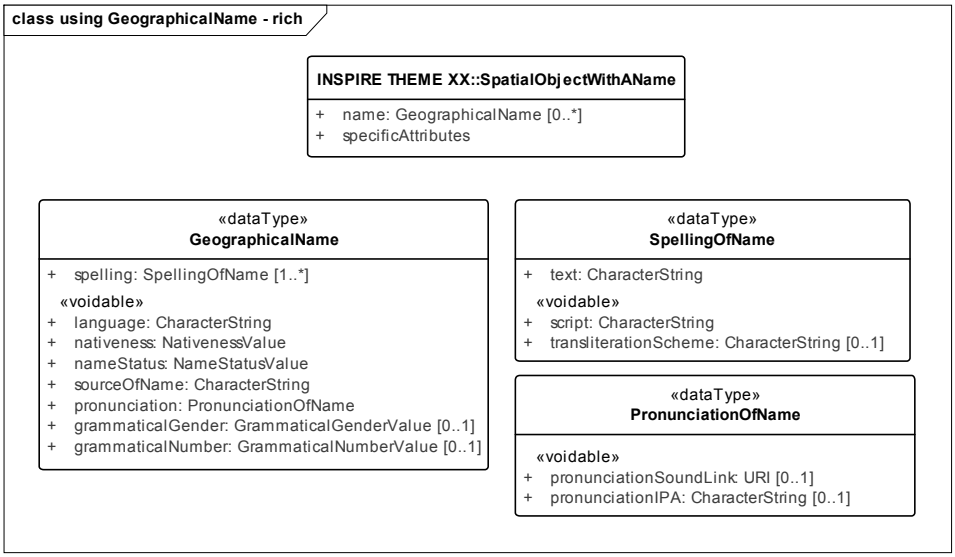
4.5 REC-06: Use alternative encoding for specific elements

Denmark supports this recommendation and hereby describes two alternative encodings for the data type GeographicalName.

Many of the themes in Annex III use the data type GeographicalName, which is defined in the Data Specification on Geographical Names ([15]). This practice is recommended in annex E, with heading "Using the datatype GeographicalName in other INSPIRE themes", of that specification: *It is recommended to use this dataType for modelling names associated to any spatial object defined in INSPIRE and holding names*.

This annex also states: *It should be noticed that the dataType 'GeographicalName' may look complex at first sight. However, when restricted to its non-voidable elements, this type is relatively simple in a context requiring managing names in multiple languages and multiple scripts. […] For the sake of simplicity,*

specifications of INSPIRE theme can however make some recommendations in their specification on how to fill the voidable elements of the dataType 'GeographicalName'. By this way, each specification may choose the adapted level of simplicity/richness of the model […] ([15, p. 92]).



Figur 1: Richest use of the data type GeographicalName ([15, p. 93], with layout modifications).

4.5.1 Encoding of GeographicalName as string

The simplest use of the data type GeographicalName is depicted in Figure 3.

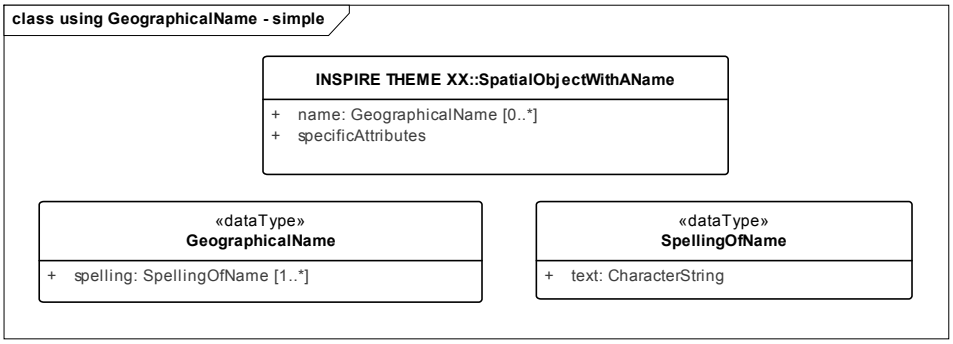


Figure 3: Simplest use of the data type GeographicalName ([15, p. 92], with layout modifications).

The simplest option for encoding this use of GeographicalName is possible when it is known that there is one spelling for names in the dataset or when it is decided to provide one spelling for a certain use case[9]. This option can be implemented by encoding GeographicalName directly as a string, without use of the INSPIRE-defined XML data type `gn:GeographicalNameType`, as is currently the case:

```
<element name="name" minOccurs="0" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element ref="gn:GeographicalName"/>
    </sequence>
  </complexType>
```

[9] The description for attribute spelling is: NOTE 1 Different spellings should only be used for names rendered in different scripts. NOTE 2 While a particular GeographicalName should only have one spelling in a given script, providing different spellings in the same script should be done through the provision of different geographical names associated with the same named place.

```
</element>
```

If also the multiplicity of the property name is flattened to a certain number (e.g. 2), using knowledge about the actual number of names in the data set, the parts of the GML application schema using this encoding would conform to compliance level SF-0 of the GML simple features profile standard (Table 1) and would e.g. look as follows:

```
<element name="name_1" minOccurs="0" maxOccurs="1" type="string" />
<element name="name_2" minOccurs="0" maxOccurs="1" type="string" />
```

Note that the above encoding is similar to what was demonstrated in [1]. The difference is that the process to come to this encoding is different, and hence, the name of the XML element is different. In the example above, the type GeographicalName is mapped directly to a string. In [1], `GeographicalName.name.spelling.text` was the only part of GeographicalName that was part of the profile and then mapped to a string, as is done with all CharacterStrings.

```
<element name="name_1.spelling.text" minOccurs="0" maxOccurs="1" type="string" />
<element name="name_2.spelling.text" minOccurs="0" maxOccurs="1" type="string" />
```

Conversion rules for decoding into a full INSPIRE application schema could be e.g. the following rules:

- name.spelling.script = "Latn”
- name.spelling.transliterationScheme is never present
- name.language is void, with reason “Withheld”
- name.nativeness is void, with reason “Unpopulated”
- name.nameStatus is void, with reason “Unpopulated”
- name.sourceOfName is void, with reason “Unpopulated”
- name.pronunciation is void, with reason “Unpopulated”
- name.grammaticalGender is void, with reason “Unpopulated”
- name.grammaticalNumber is void, with reason “Unpopulated”

4.5.2 Encoding of GeographicalName as string with language information

An important reason for using GeographicalName is to be able to provide the language of a name, as depicted in Figure 4. This can be achieved in XML by providing the `xml:lang` attribute. GML 3.3 provides an XML data type for this: `gmlxbt:LanguageStringType` ([16, p. 12]).

This encoding conforms to compliance level SF-1 of the GML simple features profile standard.

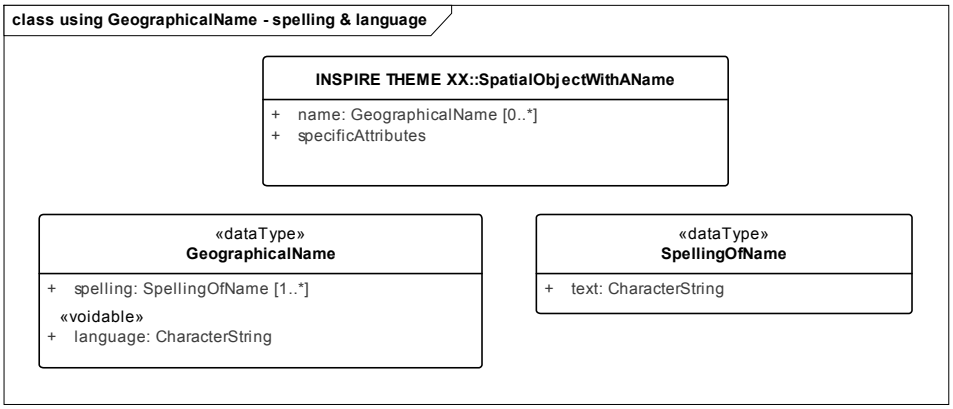


Figure 4: Second-simplest use of the data type GeographicalName.

If also the multiplicity of the property name is flattened to a certain number (e.g. 2), the GML application schema would look as follows:

```
<element name="name_1" minOccurs="0" maxOccurs="1" type=" gmlxbt:LanguageStringType" />
<element name="name_2" minOccurs="0" maxOccurs="1" type=" gmlxbt:LanguageStringType" />
```

An instance document could then look as follows:

```
< ManagementRestrictionOrRegulationZone>
  <…>
  <name_1 xml:lang="da">Nordsøen område 3</name_1>
  <name_2 xml:lang="en">North Sea area 3</name_2>
  <…>
</ ManagementRestrictionOrRegulationZone>
```

5 References

[1] AGENCY FOR DATA SUPPLY AND EFFICIENCY. “The Danish suggestion” for an approach to reduce complexity and still be compliant with the Directive (and IR DS annex 3). *3rd Meeting of the MIG-sub-group on the “fitness for purpose action 2016.1” (Reflection Group)* [online]. 2016. Available from: https://ies-svn.jrc.ec.europa.eu/attachments/download/2115/fitness%20for%20purpose%202016.1%203rd%20meeting%20danish%20suggestion.pptx

[2] GDI-DE COORDINATION OFFICE GERMANY and WETRANSFORM GMBH. *INSPIRE Fitness for Purpose - Recommendations for Changes* [online]. 2017. Available from: https://ies-svn.jrc.ec.europa.eu/attachments/download/2136/%5BDOC9%5D_170510_FitnessForPurpose_RecommendationsForChange_final.pdf

[3] ISO/TC 211. ISO 19106:2004: *Geographic information — Profiles*. 2004.

[4] INSPIRE DRAFTING TEAM “DATA SPECIFICATIONS.” *D2.7: Guidelines for the encoding of spatial data, Version 3.3* [online]. 8 April 2014. INSPIRE Drafting Team “Data Specifications.” Available from: http://inspire.ec.europa.eu/documents/guidelines-encoding-spatial-data

[5] DRAFTING TEAM “DATA SPECIFICATIONS.” *D2.5: Generic Conceptual Model, Version 3.4* [online]. 8 April 2014. Drafting Team “Data Specifications.” Available from: http://inspire.ec.europa.eu/documents/inspire-generic-conceptual-model

[6] PORTELE, Clemens (ed.). *OpenGIS Geography Markup Language (GML) Encoding Standard 3.2.1* [online]. 27 August 2007. OGC. [Accessed 18 April 2017]. Available from: http://portal.opengeospatial.org/files/?artifact_id=20509

[7] Contents of attribute nilReason (e.g. unknown vs: http://inspire.ec.europa.eu/codelist/VoidReasonValue/Unknown) : INSPIRE Thematic Clusters. [online]. [Accessed 4 June 2017]. Available from: https://themes.jrc.ec.europa.eu/discussion/view/61478/contents-of-attribute-nilreason-eg-unknown-vs-httpinspireeceuropaeucodelistvoidreasonvalueunknown

[8] DG ENV, JRC and EEA. *Action 2016.1 Fitness for purpose – addressing the challenges of interoperability for Annex III datasets* [online]. 24 May 2017. Available from: https://ies-svn.jrc.ec.europa.eu/attachments/download/2117/%5BDOC3%5D_2016.1%20Fitness%20for%20purpose%20Annex%20III.pdf

[9] INSPIRE DRAFTING TEAM “DATA SPECIFICATIONS.” *Data Specifications Template* [online]. 4 February 2013. Available from: http://inspire.ec.europa.eu/documents/data-specifications-template

[10] ISO/TC 211. ISO 19118:2011: *Geographic information — Encoding*. 2011.

[11] AMBLER, Scott W. Mapping Objects to Relational Databases: O/R Mapping In Detail. [online]. Available from: http://www.agiledata.org/essays/mappingObjects.html

[12] VANPARYS, Heidi and ECHTERHOFF, Johannes. *Mapping between UML model and replication XSD. Requirements for an automatic generation using ShapeChange*. 29 June 2015. SDFE, WorkZone: D-15364

[13] VAN DEN BRINK, Linda, PORTELE, Clemens and VRETANOS, Panagiotis A. (eds.). *Geography Markup Language (GML) simple features profile (with Corrigendum)* [online]. 5 April 2012. Available from: http://portal.opengeospatial.org/files/?artifact_id=42729

[14] THOMPSON, Henry S, BEECH, David, MALONEY, Murray and MENDELSOHN, Noah (eds.). *XML Schema Part 1: Structures Second Edition* [online]. 28 October 2004. W3C. Available from: http://www.w3.org/TR/xmlschema-1/

[15] INSPIRE THEMATIC WORKING GROUP GEOGRAPHICAL NAMES. *D2.8.I.3 Data Specification on Geographical Names – Technical Guidelines* [online]. 17 April 2014. European Commission Joint Research Centre. Available from: http://inspire.ec.europa.eu/id/document/tg/gn

[16] *OGC® Geography Markup Language (GML) — Extended schemas and encoding rules* [online]. 7 February 2012. OGC. Available from: https://portal.opengeospatial.org/files/?artifact_id=46568