TSMAD27-4.3.10B

Paper for consideration by TSMAD

Codelists

Submitted by:	SNPWG / Jeppesen
Executive Summary:	This is a proposal for adding code lists to S-100 Edition 2.0.0.
Related Documents:	(1) S-100 Ed. 1.0.0
Related Projects:	(1) S-100

1 Introduction/Background

Codelists are described in ISO 19103 as "open" enumerations. Standards in the ISO 191xx series use codelists for lists of values which depend on domain and circumstances. The ISO 191xx series of standards, GML, and the INSPIRE project guidelines make extensive use of **Codelist** data types.

ISO 19103 states that *CodeList*s can be extended during runtime. It also mentions long lists of potential values as another situation where codelists can be used. ISO 19115 (Metadata) defines several codelists, because it needs to define enumerated types whose membership is determined by domain and circumstances (e.g., distribution media).

ISO 19118 includes models of "dictionary" and "codelist." GML develops the ISO 19118 dictionary concept into an XML dictionary package that can be used for code lists. GML prescribes two different ways of encoding code lists – as an enumeration that also allows "extra" values, or using an external dictionary. GML 3.3 (OGC 10-129r1) broadens the scope of dictionary implementations to allow other current Web technologies for dictionaries..

The INSPIRE project makes extensive provisions for code lists from the modelling and application schema perspective. The INSPIRE guidelines [IN.D.2.5] recommend the use of code list for an attribute type with coded values, if the set of allowed values "may be extended by user communities or without a major revision of the data specification".

S-100 Edition 1.0.0 does not define a *CodeList* data type. § 1-4.8.1 states that code lists are to be implemented as ordinary enumeration types. On the other hand § 4a-5.1 mentions "dictionaries to implement the ISO 19115:2005 code lists". Implementations are not currently available.

This paper describes a proposal for the inclusion of code lists in S-100 Edition 2.0.0..The content is based on ISO 19103, ISO 19136, OGC 10-129r1, and the INSPIRE guidelines.

2 Terms and Abbreviations

INSPIRE Infrastructure for Spatial Information in Europe (EU project)

3 References

IN.D.2.5: D2.5: Generic Conceptual Model, Version 3.4rc3. INSPIRE draft document D2.5_v3.4rc3, 05 April 2013.
IN.D2.7: D2.7: Guidelines for the encoding of spatial data, Version 3.3rc3. INSPIRE draft document D2.7_v3.3rc3, 11 June 2013.
ISO 19103: Geographic Information – Conceptual Schema Language.
ISO 19115: Geographic information – Metadata
ISO 19118: Geographic information – Encoding
ISO 19136: Geographic Information – Geography Markup Language
OGC 10-129r1: Geographic Information – Geography Markup Language (GML) – Extended schemas and encoding rules
SKOS: SKOS – Simple Knowledge Organization System – Reference. W3C Recommendation, 2009. http://www.w3.org/TR/2009/REC-skos-reference-20090818/.

4 Discussion/Analysis

4.1 Modelling considerations

ISO 19103 states that "if all the elements of the list are known, an enumeration shall be used; if only the likely values of the elements are known, a code list shall be used." The INSPIRE guidelines [IN.D.2.5] recommend the use of code list if the set of allowed values "may be extended by user communities or without a major revision of the data specification".

S-100 products should also consider practical criteria, namely the size of the encoded list and the likelihood of its reuse in different product specifications – long lists which are likely to be reused in multiple domains are good candidates for becoming independent packages, which can be maintained as UML packages and XSD files, independently of any particular product specification and can be imported into different application schemas or XML schemas.

4.2 Extending S-100 with code lists

There are 4 ways to model/implement code lists in S-100:

A. Ordinary enumerations (as now). Edition 1.0.0 does not provide for open enumerations, i.e., the "other: …" construct for "extra" allowed values is not mentioned. This merges codelists completely into feature catalogues and is the most complex and least flexible to maintain but simplest to implement.

B. External Enumerations, implemented as ordinary enumerations but maintained separately and imported into feature catalogues. This is more flexible to maintain and distribute than ordinary enumerations but more complex to implement. This is in essence an ordinary enumeration with different technical features and maintenance and update procedures.

C. Enumeration with pattern, implemented as a union of an enumeration with a pattern in the format "other: ...". Doing anything very useful with the "extra" values (i.e., using them in portrayal rules, defining business logic around them, etc.) risks fragmentation of the base product specification into unofficial variants. There are limited circumstances in which this is useful and a use case for this option is described in Section **Erreur ! Source du renvoi introuvable.**.

D. As an external dictionary, using the GML or INSPIRE dictionary format and published as an Internet resource. This is the most capable and functional implementation but also the most complex to implement. For example, additional meta-information such as aliases can be made available to the application.

This proposal describes how Options C and D can be added to S-100. Option B is basically the same as Option A but uses some of the advanced capabilities of UML modelling tools and XML[1], and may be better covered by an "Informative" clause in S-100 or as part of a separate publication.

Product specifications should balance all relevant considerations when deciding which approaches to use. Guidance for specification authors is provided in Annex A.

4.3 Application schemas

Code lists are modelled as classes with tagged values. Code lists corresponding to Option C list the known literals as attributes. In the Option D, no attributes are listed. Figure 1 shows two examples of codelists. The Languages codelist is an example of a codelist modelled as an extensible enumeration (indicated by the tagged values *asDictionary=false* and *extensible=true*) and the Countries codelist is an example of a codelist modelled by an external dictionary (indicated by tagged value *asDictionary=true*) whose location is given by its *vocabulary* tagged

[1] Essentially, the use of separate UML packages for widely-used enumerations along with XML schemas or schema fragments, perhaps also FC and PC fragments. The fragments can be used with "import" or "include" statements, or merged into other XML schemas or documents. If such common enumerations are listed in a separate section of the GI registry they would be conveniently available to specification developers. Persistent links to the relevant XML mini-schemas, FC and PC fragments, would be provided in the same place.

value. The proposed tags are similar to those defined in GML and the INSPIRE guidelines [IN.D.2.5] except that extensibility is simplified to true/false instead of the four options allowed by INSPIRE.

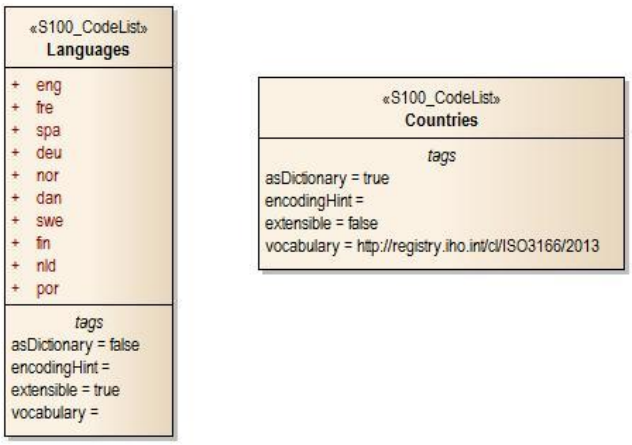

Figure 1. Two examples of CodeLists

4.4 Current Implementations

In the ISO 191xx series, code lists are defined using a Dictionary type (code-value bindings). They may be implemented as Internet resources.

Data formats may use “code list extractions” created by extracting codes or values from a codelist dictionary. The effect is to allow data formats to use either an external dictionary for code lists or convert them to ordinary enumerations for practical implementation purposes. The decision as to which alternative to use in any particular product specification should depend on the circumstances of the data product and its use environment.

INSPIRE requires that code lists be represented as dictionaries. This is the most flexible and capable implementation but will be the most complex in specification and implementation, because the specification must also specify the dictionary format and implementations must be able to utilize (access and/or parse) dictionaries.

INSPIRE codelists can be subsets of other codelists. Super-class/sub-class relationships can be used to create partitions of code lists. Code lists can be designated in INSPIRE application schemas as fixed or extensible.

4.5 Candidate code lists

ISO 693 codes for languages: Draft S-10x product specifications use ISO 639-3 (alpha-3 codes). The set of languages expected to be used in maritime information is relatively large compared to most enumerated attributes - 30-40 of the ISO language codes may be needed[2] though the number may grow to over 100 after variants, sub-national languages, and coastal and riparian states which are not IHO members are considered. Data products cannot necessarily be restricted to a small number of languages - English can be the mandatory language for crucial information such as ENCs but other publications and local information are likely to be published only in other languages (and data products will contain extracts from these). Changes to the list of codes are expected to be rare. The set of language codes will certainly be shared by most S-10x product specifications. Official lists are maintained by registration authorities (SIL International for ISO 639-3).

ISO 3166 Country codes: There are currently 249 officially assigned codes. The number or country codes used in maritime information will be comparable to the number for language codes, for much the same reasons, but will probably be lower though not significantly so. ISO 3166-1 defines alpha-2, alpha-3, and 3-digit numeric codes. ISO provides the alpha-2 codes free, in text, online, and XML forms. ISO 3166-2 defines codes for the names of principal subdivisions (provinces/states). Current drafts of S-10x product specifications use the alpha-2 code.

[2] ISO 639-3 (the standard mentioned in current S-1xx documents) is intended to include all natural languages. The official list of has over 7000 language codes. The number of languages likely to be used in maritime information is closer to the number of coastal states. IHO currently has 80 member states.

NGA World Port Index: This lists over 4000 ports, shipping facilities, and oil terminals throughout the world. It gives the country, location, characteristics, known facilities, and available services of ports selected based on criteria established by the NGA.

Radio channels and frequency assignments: Radio communications information includes frequency assignments to specific channels defined by the International Telecommunications Union (ITU) and consists of long collections of codes (channel designators), transmitting/receiving frequencies, and permitted uses. Tables of maritime communications information include VHF, MF, HF channels. Entries describing channels are often annotated with additional notes on geographic areas, regional variations, availability, etc.

4.6 Data formats

Specification authors should note that data formats can be distinguished from the normative model yet synchronized with it, e.g., it is possible to transform an extract from a dictionary (option **Erreur ! Source du renvoi introuvable.**) into an XML fragment which is merged into a feature catalogue which treats the attribute as an ordinary enumeration (option **Erreur ! Source du renvoi introuvable.**), obviously an additional procedure is involved for future maintenance of the FC and product specification, which procedure is specific to that product specification.

4.6.1 GML and other XML data formats

The normative format is determined by the tagged value *asDictionary* attached to the corresponding UML class in the application schema. GML 3.3 (OGC 10-129r1) amends GML 3.2 (ISO 19136) to allow "any suitable syntax or encoding" for the external list, and suggests that contemporary Web technologies including semantic web representations be considered. The normative formats for the two CodeList options are:

Option C, Enumeration with pattern: Encode in conformance to ISO 19136 Clause E.2.4.9 as a union of an enumeration and a pattern of the form "other: \w{2,}" that allows for text values prefixed with "other: ". For example, assuming a codelist which explicitly lists "Norwegian" but not Nynorsk and Bokmål:

```
<language>nor</language>          <!-- Norwegian is an explicitly enumerated value -->
<language>other: nno<language>    <!-- Norwegian Nynorsk is not an enumerated value -->
```

Option D: External Dictionary: Encode in the instance document as a reference to the dictionary entry and define the dictionary (vocabulary) in any standard dictionary format (ISO and W3C define formats). Assuming the IHO maintains a dictionary of ENC aggregated features (with code 1 corresponding to "Range System"), data products could refer to the range system thus:

```
<categoryOfAggrFeature xlink:href="http://registry.iho.int/cl/s101/aggr/ver03/1"/>
```

Dictionary version information can be indicated ("ver03" in the example). The dictionary must be available on the Internet and may be included with distributed software. A format for the dictionary is not being specified at this time, for the same reasons as OGC 10-129r1 (no clearly dominant format at present).

4.6.2 ISO 8211 encodings

Option C, Enumeration with pattern: To accommodate producer-defined values ("other: xyz") this can be encoded either as a "text" type (character string) or as a complex attribute with an integer sub-attribute (for the listed allowed values) and a text sub-attribute (the "other:..." values).

Option D: External Dictionary: This can be encoded in two ways:

1) A URI data type with value a URI constructed by combining the URI for the vocabulary (dictionary) and the item code. E.g., http://registry.iho.int/codelist/s23/1953/1 for the Baltic Sea (item 1 in the 1953 edition of IHO publication S23 – Limits of Oceans and Seas).

2) A complex attribute with two sub-attributes: Vocabulary location (URI) and item code (text). To use the same example: sub-attributes are *vocabulary*= http://registry.iho.int/codelist/s23/1953/ and *itemCode*=1.

Method 1 is recommended for Option D, as it reduces data complexity. No change to the ISO 8211 format is needed, but obviously the attribute will have to be encoded as a string-valued attribute instead of a numeric-valued attribute.

4.7 Dictionary formats

The reasonably mature formats for dictionaries are currently GML [ISO 19136, OGC 10-129r1], and SKOS [SKOS]; RDF and OWL are less preferable possibilities. This paper recommends use of GML or SKOS formats.

4.8 Registry, Distribution, and Maintenance

A code list register would be useful for facilitating reuse of codelists in product specifications as well as applications.

Codelists are generally maintained by a central responsible body. The maintenance of codelists should follow normal GI registry procedures including versioning, change control, etc. Codelists could be maintained by a domain expert group just like product specifications, and their other treatment and metadata in the GI registry can be similar to product specifications, except that the artefacts involved are basically just parts of a product specification - fragments of application schemas, feature catalogues, and XML schemas, and perhaps a fragment or templates for portrayal catalogues. We suggest the GI registry should treat them like product specifications with the unnecessary clauses and metadata omitted. Metadata for citation of the parent source (e.g., ISO standards), lineage, versioning, representation language, official Internet URIs of the vocabulary, XML schemas, etc., is obviously required.

Applications may download code lists but applications using only a local copy are susceptible to content changes and divergence – in the maritime domain this means a maintenance/distribution regime is needed and deletions from external code lists may need to be limited or linked to new versions of product specifications.

5 Recommendations

Recommendation 1: Add a CodeList type to S-100.

Recommendation 2: Add a codelists register to the GI registry, structured like the product specifications register but omitting the components and metadata elements not applicable to codelists.

6 Justification

Allowing codelist types in S-100 will provide product specification developers with the flexibility to design data products for the constraints prevailing in different application domains, including distribution and maintenance considerations as well as implementation. Codelists will also facilitate reuse of data models.

7 Impacts

Product specifications currently being prepared need not change unless specification writers desire to implement one of the “open form” implementations described in this paper.

Implementers of specifications which use a “dictionary” data format will need to adapt implementations to implement CodeLists data type and lookup items in the named vocabularies. Implementers of specifications which provide an enumerated data format will not need to change. Specification authors will need to develop translation tables between different data formats.

8 Conclusion

Codelists provide a way to model open, flexible enumerations. They provide specification authors with the capability to design specifications with more flexible distribution and maintenance regimes and facilitate reuse of the work of external organizations and reuse across domains. Current standards provide two methods of modelling and implementing them. This paper recommends addition of both methods to S-100 Edition 2.0.0 and also the addition of a codelists register to the IHO GI registry to facilitate reuse.

9 Actions Requested

TSMAD is requested to:

- Add codelist types to S-100; if agreed, consider the further actions below:
- Adopt both Options C (enumeration with pattern) and Option D (external dictionary) as representations of codelist types.

- Review and amend the changes suggested in the accompanying change proposal form and include the finalized changes in S-100 Edition 2.0.0;
- Add a codelists register in the IHO GI registry.

Annex A. Guidance for product specification authors

Product specifications should balance all relevant considerations, e.g., implementation costs, application operational environment, cross-domain reuse, and reduction of maintenance and distribution efforts, when deciding which approach to use for any particular attribute.

A.1. Modelling

When deciding between using a codelist and enumeration, consider the completeness, stability, source, reuse, and application dependencies of the list of values.

- If the set of allowed values is fixed and reasonably short (say, fewer than 20 values?), an enumeration must be used.
- If the list is fixed but long, an enumeration is preferred but a "dictionary model" codelist may be used.
- If only the likely values of an enumeration are known, or the list may be extended by data producers or the user community, a codelist must be used. Whether the "dictionary" or "open" form is preferable depends on who might add values – if it is maintained by an organization, the dictionary form is preferable, if user communities or data producers may add values, the "open" form is preferable.
- If the allowed values change frequently and the list should be updated without major revisions of the product specification, a codelist may be used. The "dictionary" form may be preferable under these circumstances.
- If application logic or portrayal rules depend on values, an enumeration is preferred but a codelist may be used if the logic/rules can be written to cover all possible values (e.g., using wildcards or defaults), or otherwise allow graceful recovery from unanticipated values.
- Collections which have internal structure (e.g., types and subtypes of vessels) should be modelled as "dictionary" codelists, pending discussion of the matter by ISO TC211.

A.1.1. Hierarchies of codelists

A code list may also be used as a super-type for more specific code lists. The vocabulary of the super-type is the union of the vocabularies of its sub-types[3]. If additional values are permitted the super-type must have tag *extensible=true*, otherwise it must have *extensible=false*. Practically, this allows vocabularies developed by different domain expert groups or organizations to be merged.

A.2. Codelists maintained by external organizations

If there is an existing well-established codelist maintained by a responsible source, it can be referenced in an application schema. The code list should meet the following requirements[4]:

- It must be managed by a responsible source – an official national or international standards body, long-established user community, group, or consortium.
- The codelist and its values must be identified by persistent HTTP URIs.
- The list should be well-maintained i.e. all its values must remain available forever, even if they have been deprecated, retired or superseded.
- The list should be in a dictionary language accepted for use in S-10x product specifications.

The IHO may be requested to arrange for the translation, reproduction, and maintenance of codelists meeting only some of the above requirements. Note that this may necessitate a discussion between the IHO and the source.

[3] Note that the super-type cannot augment the union set with additional definitions. This conforms to the INSPIRE usage but may be worth reconsidered if an argument for such augmentation is made by OEMs, TSMAD, or SNPWG.

[4] Adapted from reference IN.D2.5.

A.3. Data formats of codelist typed attributes

The codelist model in S-100 is designed to be flexible by decoupling application schema from data format to some extent. Data formats may use "code list extractions" created by extracting codes or values from a codelist dictionary and treat them as ordinary enumerations. The effect is to allow data formats to use either an external dictionary or ordinary enumerations. For example, an XML data format might convert an *ISO3166CountryCodes* codelist maintained by IHO into an XML Schema type:

```
<xs:simpleType name="ISO3166CountryCodesType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EN"/>
    <xs:enumeration value="FR"/>
    ... other country codes ...
```

As far as implementations using that schema are concerned, it is indistinguishable from an ordinary enumeration. The decision as to which alternative(s) to use in any particular product specification should depend on the circumstances of the data product and its use environment. The decision should be made by the product specification authors when developing the data format. Obviously allowing different data formats to use different representations introduces additional maintenance requirements relating to some data formats, these would be limited to the formats which use "closed" representations (i.e., convert the codelist to an ordinary enumeration).

A.3.1. GML and other XML data formats

Enumeration with pattern: The data format in XML schemas must conform to ISO 19136 E.2.4.9, i.e., a union of an enumeration and a pattern of the form "other: \w{2,}".

Examples of use (assuming a codelist which explicitly lists "Norwegian" but not Nynorsk and Bokmål):

```
<language>nor</language>          <!-- Norwegian is an explicitly enumerated value -->
<language>other: nno<language>    <!-- Norwegian Nynorsk is not an enumerated value -->
```

External Dictionary: The data format in XML schemas must be the XML Schema built-in types *anyURI*. The use of spaces is discouraged.

Example:

In XML schema: Type definition: `<xs:simpleType name="namedSeaType" type="xs:anyURI">` and later (in feature definition): `<xs:element name="namedSea" type="namedSeaType"/>`

In dataset: `<namedSea xlink:href="http://registry.iho.int/cl/s23/1953/1"/>`

A.3.2. ISO 8211 encodings

Enumeration with pattern: To accommodate producer-defined values ("other: xyz") this can be encoded either as a "text" type (character string) or as a complex attribute with an integer sub-attribute (for the listed allowed values) and a text sub-attribute (the "other:..." values).

External Dictionary: This can be encoded in two ways:

1. A URI data type with value a URI constructed by combining the URI for the vocabulary (dictionary) and the item code. E.g., `http://registry.iho.int/codelist/s23/1953/1` for the Baltic Sea (item 1 in the 1953 edition of IHO publication S23 – Limits of Oceans and Seas).
2. A complex attribute with two sub-attributes: Vocabulary location (URI) and item code (text). To use the same example: sub-attributes are *vocabulary*= `http://registry.iho.int/codelist/s23/1953/` and *itemCode*=1.

The first method is recommended as it reduces data complexity.

A.4. Dictionary formats

Use of GML dictionary or SKOS format is recommended. Other formats may be considered under compelling circumstances or after the development of standards in ISO or elsewhere.