

Paper for consideration by TSMAD
S-100 Codelists Progress

| | |
|---------------------------|--|
| Submitted by: | Jeppesen |
| Executive Summary: | This paper provides an update on code lists for S-100. |
| Related Documents: | (1) S-100 Ed. 1.0.0 |
| Related Projects: | (1) S-100 |

1 Introduction/Background

TSMAD 25 assigned an action to produce necessary text and UML diagrams to provide for the inclusion of code lists in S-100. This paper reports on progress of this item.

2 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 3.3rc2. INSPIRE draft document D2.7_v3.3rc2, 15 June 2012.

ISO 19103: Geographic Information – Conceptual Schema Language.

ISO 19115: Geographic information – Metadata

ISO 19118: Geographic information – Encoding

ISO 19136: Geographic Information – Geography Markup Language

3 Discussion/Analysis

Work on this action included a review of the relevant standards from the ISO 191xx series and guidelines prepared by the INSPIRE project. The following is a summary of the results.

The ISO 191xx series of standards, GML, and INSPIRE make extensive use of **Codelist** data types. S-100 Edition 1.0.0 does not use codelists – it implements code lists as enumeration types according to § 1-4.8.1. On the other hand § 4a-5.1 seemingly contradicts this by mentioning that implementation of metadata includes “XML documents containing dictionaries to implement the ISO 19115:2005 code lists (XML data dictionaries of the ISO 19115:2003 code lists in GML format)” and “XML data dictionaries of the S-100 Geographic Extent Names and Search Words”.

ISO 19103, 19115, and 19118

ISO 19103 defines the CodeList type as describing an open enumeration that needs to be represented in such a way that it can be extended during runtime. It states that “Code lists are useful for expressing a long list of potential values. 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.”

ISO 19115 defines several codelists, because it needs to define enumerated types whose membership is determined by domain and circumstances (e.g., distribution media).

Code lists are defined using a Dictionary type (code-value bindings). They may be implemented as Internet resources. Encodings may use “code list extractions” created by extracting codes or values from a codelist dictionary. The effect is to allow encodings to use either an external dictionary for code lists or use values as for enumerations.

GML (ISO 19136)

GML (ISO 19136) defines a “dictionary” package that is intended for encoding dictionaries of code lists. GML provides two different ways of encoding code lists in GML XSD files (controlled by a tagged value for the UML class in the application schema):

1. An enumeration that also allows “extra” values in the format “**other: xyz**” where the “xyz” may be any string.
2. A dictionary in the GML dictionary format.

INSPIRE

INSPIRE makes extensive provisions for code lists. 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*” (emphasis added).

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 as fixed or extensible. 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 should be able to use dictionaries.

Extending S-100 with code lists

There are 3 ways to model/implement code lists:

- 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. Special Enumerations, implemented as enumerations but maintained separately and imported into feature catalogues. This is more flexible to maintain but more complex to implement.
- C. 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.

4 Other considerations

A code list register would be useful for product specifications as well as applications.

Codelists are generally maintained by a central responsible body. They are not managed as part of application schemas. The maintenance of codelists should follow normal procedures including versioning, change control, etc.

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.

Code lists are being discussed with SNPWG and more will be known about use cases following those discussions.

5 Conclusion

This paper summarizes the state of codelists in the relevant ISO standards and the approach to codelists in the INSPIRE framework. Much work remains to be done, including an assessment of the costs and benefits of extending S-100 with codelists. TSMAD is invited to discuss the ideas summarized here and continue the action item to TSMAD 27.

6 Actions Requested

TSMAD is invited to:

- note this paper
- continue the action item to TSMAD 27