Formalisation of Code Lists and Their Values – The Case of ISO 19152 Land Administration Domain Model

Abdullah KARA, Alexandra ROWLAND, Peter VAN OOSTEROM, The Netherlands, Erik STUBKJÆR, Denmark, Volkan ÇAĞDAŞ, Turkey, Erwin FOLMER, Christiaan LEMMEN, Wilko QUAK, The Netherlands and Laura MEGGIOLARO, Italy

Key words: Code list, Formalisation, Metamodel, Semantic web, ISO 19152, LADM.

SUMMARY

A code list in Unified Modeling Language (UML), a simple list of values without any structure, can be employed as a simple data type to further capture the semantics of a domain. The code list values as used in international standards (e.g. ISO and OGC) are generally presented without definition, reference to the source of a definition, multi-lingual alternative term support and semantic relationships (e.g. hierarchical, associative). Moreover, managing, implementing, and maintaining UML code lists can be considered as a difficult task since they generally do not provide structured and semantically enriched values. This is also true in the case of ISO 19152:2012 Land Administration Domain Model (LADM), which is currently under systematic review and adding more content, meaning and structure to code list values could be considered an improvement.

In last decade, there is a growing interest in representing terms as well as code list values using Semantic Web technologies (e.g. RDF, OWL, SKOS, SPARQL) and making them available on a registry (e.g. ISO/TC211 Geolexica, OGC Definitions Server, INSPIRE code list register, BARTOC), including land administration domain (e.g. CaLAThe, LandVoc). However, there is no joint understanding in structuring, extending and maintaining code list values, which may be achieved through an agreed metamodel. Such a metamodel should also provide insight into content, localisation (multi-lingual support), versioning and implementation.

The aim of this study is to propose a framework (basically a metamodel) for structuring, extending, maintaining and implementing semantically enriched code lists, and to discuss the application of the proposed framework to be included in the revision of LADM. To achieve this aim, the requirements for refined code list values are firstly collected considering existing thesauri, vocabularies and standards. Subsequently, a metamodel is proposed for the refined code list. The proposed metamodel is applied to a code list of LADM as well as a part of a selected country profile.

Formalisation of Code Lists and Their Values – The Case of ISO 19152 Land Administration Domain Model

Abdullah KARA, Alexandra ROWLAND, Peter VAN OOSTEROM, The Netherlands, Erik STUBKJÆR, Denmark, Volkan ÇAĞDAŞ, Turkey, Erwin FOLMER, Christiaan LEMMEN, Wilko QUAK, The Netherlands and Laura MEGGIOLARO, Italy

1. INTRODUCTION

ISO TC/211 and OGC standards generally uses the class diagram (static view modelling) of the UML to describe its schema which basically consists of classes, their attributes and relationships among classes. In this way of modelling, data types are assigned to attributes in order to specify the allowable characteristics of them. An existing data type (i.e. previously defined by another standard) or a new one defined specifically for a new model can be utilised to define the nature of an attribute. One of the options to determine which values an attribute can have is to use code lists.

A code list is a '*defined set of valid values of an attribute parameter*' (cf. ISO 28258:2013). Code lists can be considered as a supplement to the classes and associations within a standard. They are included, because for various reasons it was not possible to specify a complete list within a standard, just initial examples of code list values are provided. It is important to note that the code list values provide a basis for further structuring of the domain, as it can be stated that when modelling for actual systems to be implemented typically half of the efforts are devoted to the creation of the UML class diagrams as backbone of the structure. The other half of the efforts are spent on defining the actual code list values, and contain many domain semantics. However, the code lists as used in international standards (e.g. ISO and OGC) are generally just mentioned in the UML class diagrams and present only a simple list of example values without any structure, (in many cases) therefore without definition, reference to the source of a definition, multi-lingual alternative term support and semantic relationships (e.g. hierarchical, associative). This is also true in the case of ISO 19152:2012 LADM with the code list values defined in an informative Annex J.

LADM, a reference model providing '*an extensible basis for the development and refinement of efficient and effective land administration systems, based on a Model Driven Architecture (MDA)*', specifies a number of code lists (e.g. LA_PartyType, LA_MortgageType, etc.) in the informative part of the standard. These code lists only include example values without any definition, alternative term, and semantic relationship (e.g. hierarchical, associative), and the specification of them is left to '*User communities [who] have to define and manage their own values when implementing*' the standard. LADM is currently under the systematic review of ISO and adding more content, meaning and structure to its code list values has been considered to take a step forward in development of LADM. Paasch et al. (2015) and Stubkjær et al. (2018) propose code lists as a mean of internationalisation by which the classes of the LADM may be related to the jurisdiction concerned.

In recent years, there is growing interest in representing terms belonging to a particular domain as well as code list values using Semantic Web technologies (e.g. RDF, OWL, SKOS, SPARQL) and making them available on a registry (e.g. the ISO/TC 211 Multi-Lingual Glossary of Terms (MLGT), alias Geolexica[1], OGC Definitions Server[2], INSPIRE code list register[3], BARTOC[4]) on the Web. This interest is also the case in the domain of land administration.

The Cadastre and Land Administration Thesaurus (CaLAThe[5]), which includes terms of LADM, OGC's LandInfra, and several existing thesauri (e.g. GEMET, AGROVOC, and STW Thesaurus for Economics) and semantic relationships between the terms (Çağdaş and Stubkjær, 2015). CaLAThe is based on the Simple Knowledge Organization System (SKOS) and Dublin Core terms (properties) and is structured in Resource Description Framework (RDF) format. In order to support international joint code list management through a controlled domain vocabulary, the most recent version of CaLAThe also includes code list names and values of ISO LADM and OGC LandInfra that is made available through BARTOC[6], and also through the OGC Definitions Server [7] (Stubkjær and Çağdaş, 2021; Çağdaş et al., 2021). Another land administration related thesaurus is LandVoc[8], which covers concepts related to land governance. It is designed as a part of FAO's AGROVOC Linked Open Data set and is maintained by the Land Portal Foundation[9]. LandVoc is available in a multitude of language and is aligned with several other thesauri and vocabularies. It is basically based on RDF, SKOS and Dublin Core terms and can be consumed through AGROVOC SPARQL endpoint[10], SKOMOS and RDF format. It can be noted that LandVoc does not provide a specific solution for code list terms. Furthermore, the UML code list values defined in various INSPIRE data themes including cadastral parcel and administrative unit were structured in RDF and XML formats and made available on the INSPIRE code list register. The RDF representation of INSPIRE code list values do not include hierarchical relationships between code list values, and references to external code list values. For a more detailed comparison of the knowledge organisation systems related the land and geospatial domains can be found in Çağdaş et al. (2021).

It is worth noting here that code list values can be structured in a number of ways within the framework of the Semantic Web, using various formalisms, including RDF. The differences related to content and utilised vocabularies (e.g. SKOS, Dublin Core, FOAF) in the land administration thesauri can be a good example to that. More specifically, the definition of concepts (terms) is specified by *rdf:Description* and *skos:definition* properties in the LandVoc

[1] https://isotc211.geolexica.org/
[2] https://www.ogc.org/def-server#
[3] https://inspire.ec.europa.eu/codelist/
[4] https://bartoc.org/
[5] http://cadastralvocabulary.org/
[6] https://bartoc-skosmos.unibas.ch/CaLAThe/en/
[7] https://www.opengis.net/def/CaLAThe/4.0/
[8] https://www.landvoc.org/
[9] https://www.landportal.org/
[10] https://agrovoc.fao.org/sparql/

and CaLAThe, respectively. This situation may create confusions in international code list management. To overcome this issue, a joint understanding in structuring, managing, extending and maintaining code list values is needed which may be achieved through an agreed metamodel.

This study aims at presenting a framework (mainly a metamodel) for structuring, extending, registering, maintaining, and implementing semantically enriched code list values, and discusses the application of the proposed framework to LADM code lists and a related country profile. To achieve this aim, the requirements for semantically refined code list values (e.g. definition, source of definition, versioning, alternative terms, semantic relationships, languages, localisation, etc.) are firstly collected in Section 2. Next, relevant thesauri and registries for structuring and managing code list values are identified and evaluated against the earlier stated requirements. Using the outputs of the requirement analysis, a metamodel is proposed for the refined code list. The metamodel is applied to LA_MortgageType code list of LADM as well as a part of a selected country profile. Section 4 concludes the current paper.

2. REQUIREMENTS FOR A SEMANTICALLY ENRICHED CODE LIST VALUES

When developing a new model, it is important to reuse existing standards as a foundation and to continue from that point to ensure interoperability in the domain (van Oosterom et al., 2018). Since the main aim of this paper is to create a metamodel for structuring semantically enriched code lists, ISO 19150-2:2015 – Rules for developing ontologies in the Web Ontology Language (OWL) standard, which '*defines the conversion of the UML static view modeling elements used in the ISO geographic information standards into OWL*', should be firstly investigated.

ISO 19150-2:2015 uses the code list definition of ISO 19103, which is '*a datatype defining an open list of likely mnemonic identifiers. It expresses a list of potential values*'. This standard indicates that '*SKOS has been broadly adopted for vocabulary formalization. SKOS supports the codelist requirements of membership and extensibility.*' This statement is well aligned with the brief overview about land administration related thesauri given in the previous section. ISO 19150-2:2015 sets a number of requirements for the description of code lists with Semantic Web technologies which as follow:

- '*A CODELIST shall correspond to a Class , a ConceptScheme , and a Collection.*'
- '*The Class <OWL> shall be a subclass of skos:Concept.*'
- '*The SKOS concept scheme shall be related to the Class using a dct:isFormatOf property.*'
- '*Each member of the CODELIST shall correspond to an individual whose type is the Class <OWL> corresponding to the CODELIST, and with a skos:inScheme property whose value is the ConceptScheme <SKOS> corresponding to the CODELIST.*'
- *'Each member of the CODELIST shall also be member of the Collection using a skos:members declaration.*'

- ’*Each of the resources shall be annotated with the following: a label, using rdfs:label, a source for the definition, using rdfs:isDefinedBy for the IRI of the resource.*’

ISO 19150-2:2015 also specifies a requirement for extending the code list with a new value: ‘*If a CODELIST is extended with additional items that were not identified in the original model, the URI denoting a new member shall be in a URI domain appropriate to its governance. This should not be the same as the URI domain for the initial members unless the new members are defined by the same authority*.’ Moreover, this standard recommends that the use of *skos:broader*, *skos:narrower*, and *skos:related* properties in order to specify the hierarchical relationships values of a code list. In addition, the utilisation of *skos:broadMatch*, *skos:closeMatch*, *skos:exactMatch*, *skos:narrowMatch* and *skos:relatedMatch* properties are suggested to record the relationships between values of a code list and values of other concept schemes. It should be noted that the focus of ISO 19150-2:2015 is the conversion from UML to OWL. It does not include a number of important requirements for semantically enriched code lists, such as versioning, localisation, alternative values, maintenance, registration and publishing.

After several meetings with experts, thirteen requirements for semantically refined code list values are firstly identified, see first column in Table 1. Next, the selected thesauri and registries (see first row in Table 1) for structuring and managing code list values are evaluated against the identified requirements.

Before explaining the Table 1, it should be noted that CaLAThe and LandVoc are designed as thesauri in the first place, consisting of the selected terms. Some of the ISO LADM and OGC LandInfra code lists are also included in the version 4 of CaLAThe, as demonstrated through the BARTOC SKOMOS Browser[11]. Moreover, all code lists are now included with a separate CaLATheCodeList concept scheme in in the version 5 of CaLAThe (http://www.cadastralvocabulary.org/CaLAThe.rdf). Therefore, the solutions of LandVoc is given considering its thesauri structure in Table 1, while the solutions of CaLAThe is given according to its SKOS representation created for the code lists. The Geolexica and OGC Definitions Server are only evaluated with yes or no in Table 1 as they are just platforms (registries) to serve existing terms of thesauri and standards.

The selection of suitable code list values may be the first and most important aspect to be decided as it specifies the content of the schema. CaLAThe thesaurus includes almost 250 terms, extracted from LADM, OGC LandInfra, AGROVOC, GEMET and so on (Stubkjær and Çağdaş, 2021). LandVoc builds on existing land glossaries, such as the FAO’s Land Tenure Thesaurus (Ciparisse, 2003), LADM and the Global Land Indicators Initiative glossary (Çağdaş et al., 2021); while the INSPIRE code list register is designed for only serving the INSPIRE data themes. The Geolexica serves the terms given in the terms and definitions sections of ISO standards. The OGC Definitions Server makes available the terms the OGC defines or that communities ask the OGC to host on their behalf (https://www.ogc.org/def-server#).

[11] https://bartoc-skosmos.unibas.ch/CaLAThe/en/

The refined code lists and their values shall have unique identifiers, preferably with persistent URIs. It is necessary and expected to choose a URI pattern suitable for the needs. The identifiers of the code lists and their values should be represented with a property of a thesauri or a dictionary. For example, CaLAThe uses *skos:ConceptScheme* for representing all the code lists and *skos:Concept* for representing code list names and values, while LandVoc and INSPIRE uses rdf:Description property for representing all its terms (see, Table 1, R2). As can be seen here, the thesauri use different approaches to structure their RDF documents.

A preferred label should be assigned to the URI of code list values in order to make it easy to use. The *skos:prefLabel* is selected for this purpose by CaLAThe and INSPIRE, while LandVoc also utilises the *skosxl:prefLabel*. Code list values may have alternative labels and for this purpose *skos:prefLabel* and *skosxl:altlabel* are used as can be seen in Table 1.

Each code list value should have a definition preferably from a well-accepted source. CaLAThe and INSPIRE utilises *skos:definition* property to record the definition and source of the definition, while LandVoc uses *rdf:Description*.

Each code list value shall belong to at least one code list. Providing this requirement may facilitate the search of code list values and discovering cross code lists relationships. The *skos:inScheme* property is used by CaLAThe and INSPIRE to meet this requirement. Since LandVoc is not designed to represent refined code list values, it does not cover this requirement.

One of the advantages of using Semantic Web technologies is that different code list values can be linked using hierarchical and associative properties of existing thesauri. A refined code list should contain as many hierarchical, associative and mapping relationships as possible. *skos:broader*, *skos:narrower* and *skos:related* properties are utilised by CaLAThe and LandVoc in order to capture hierarchical relationships between concepts. Hierarchical relationships between code list values are not covered within INSPIRE RDF documents. Furthermore, Semantic Web technologies also enable to define mapping relationships across different code lists' values. For this purpose, CaLAThe and LandVoc use the following properties: *skos:exactMatch*, *skos:closeMatch*, *skos:broadMatch* and so on.

In order to manage the temporal aspects, refined code lists and each of their values should be versioned with begin life span, end life span, begin life span version and end life span version dates. Moreover, the procedures (i.e. adding a new value, changing a value, deleting a value, changing the definition of a value, so on) for updating code lists and their values should be clearly defined. It should be specified when a new version of a code list is released, and when a new version of a collection of code lists (e.g. LADM code lists) is released. To the best of the authors knowledge, only CaLAThe has been publishing new versions, mostly for adding more content.

Table 1. Requirements for a refined code list and solutions of existing thesauri and registries

Requirements \ Solutions	**S1. CaLAThe**	**S2. LandVoc**	**S3. INSPIRE code list register**	**S4. Geolexica**	**S5. OGC Definitions Server**
R1. Selection of code list values	OGC LandInfra, LADM, …	FAO's Land Tenure Thesaurus, LADM, …	INSPIRE Data Theme	ISO standards	Terms defined by OGC or communities
R2. Identifier for code lists and their values	skos:ConceptScheme, skos:Concept, rdf:about	rdf:Description, rdf:about	rdf:Description, rdf:about	Yes	Yes
R3. Label (preferred and alternative) for code list values	skos:prefLabel, skos:altlabel	skos:prefLabel, skosxl:prefLabel, skos:altlabel, skosxl:altlabel	skos:prefLabel, skos:altLabel	Yes	Yes
R4. Definition of code list values (and source of definition)	skos:definition	rdf:Description rdf:value	skos:definition	Yes	Yes
R5. Determining the values that belong to a code list	skos:inScheme, skos:broader	No	skos:inScheme	-	-
R6. Hierarchical and associative relationships between code list values	skos:broader, skos:narrower, skos:related	skos:narrower, skos:broader, skos:related	No	Yes	Yes
R.7 Mapping relationships across different code list values	skos:exactMatch skos:closeMatch	skos:exactMatch skos:closeMatch skos:broadMatch	No	Yes	Yes
R8. Versioning of code lists and their values (and concurrent versioning)	Partly yes	No	No	Partly yes	Partly yes
R9. Procedures for updating code list values (e.g. new, changed, deleted)	No	No	No	No	No
R10. Semantic web view of code lists	SKOS RDF/XML	SKOS RDF/XML, SPARQL Endpoint	RDF/XML	Yes	Yes
R11. Support for multiple languages (alphabets)	Yes, with labels	Yes, labels and skosxl:literalForm	Yes	Yes	Yes
R12. Support for national extension	Partly yes	Partly yes	Partly yes	-	-
R13. Selecting possible registries to publish and maintain code list values	Yes, through OGC Definitions Server, SKOMOS	AGROVOC	-	-	-

Different semantic web technologies should be used to increase the use and discoverability of code lists and their values. LandVoc has an advantage in this regard since a variety of formats is offered, including SPARQL endpoint.

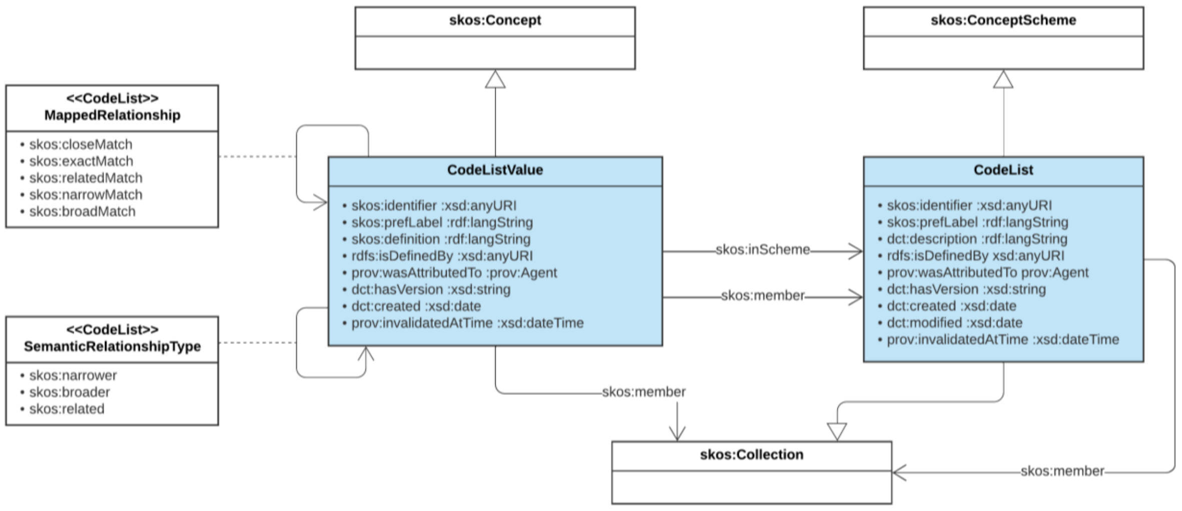
A methodology shall be defined to support for national extensions of refined code list values. CaLAThe uses labels and two letter country codes to cover different languages (only Danish and Turkish terms are covered so far), while LandVoc utilises labels, country codes and *skosxl:literalForm* property (several languages). On the other hand, there is no published methodology for supporting extensions of refined code list values.

Lastly, in order to exploit the potential of refined code list values, possible registries to maintain code list values (e.g. OGC definition server, ISO TC/211 registries) should be selected. CaLAThe was made available through OGC Definitions Server and BARTOC SKOMOS, while LandVoc can be achieved via AGROVOC. It is noted that the availability of code lists in registries facilitates the management of national code lists (Stubkjær and Çağdaş, 2021).

3. A METAMODEL FOR REFINED CODE LISTS AND THEIR VALUES

The requirements for structuring semantically enriched code list values and the solutions provided by existing thesauri and registries were investigated in the previous Section. The obtained knowledge is now used to create the code list metamodel.

The metamodel is mainly based on the requirements specified in ISO 19150-2:2015 and Table 1. The metamodel is presented in Figure 1, which illustrates the proposed structure for the creation, structuring and maintenance of the semantically enriched code lists associated with LADM.



skos:Concept
skos:ConceptScheme
<<CodeList>>
MappedRelationship
• skos:closeMatch
• skos:exactMatch
• skos:relatedMatch
• skos:narrowMatch
• skos:broadMatch
CodeListValue
• skos:identifier :xsd:anyURI
• skos:prefLabel :rdf:langString
• skos:definition :rdf:langString
• rdfs:isDefinedBy :xsd:anyURI
• prov:wasAttributedTo :prov:Agent
• dct:hasVersion :xsd:string
• dct:created :xsd:date
• prov:invalidatedAtTime :xsd:dateTime
CodeList
• skos:identifier :xsd:anyURI
• skos:prefLabel :rdf:langString
• dct:description :rdf:langString
• rdfs:isDefinedBy xsd:anyURI
• prov:wasAttributedTo prov:Agent
• dct:hasVersion :xsd:string
• dct:created :xsd:date
• dct:modified :xsd:date
• prov:invalidatedAtTime :xsd:dateTime
skos:inScheme
skos:member
<<CodeList>>
SemanticRelationshipType
• skos:narrower
• skos:broader
• skos:related
skos:member
skos:Collection
skos:member

Figure 1. Refined code list metamodel

The structure defined above makes primary use of SKOS notation which serves to represent LADM code lists as *skos:ConceptScheme* and *skos:Collection* based on the collection of code list values, here defined as *skos:Concept*. In order to provide a complete overview of the details contained within this representation, each part of this metamodel will be discussed and, in order to support this discussion, implementation examples of each part of the metamodel are provided below.

3.1.1 LADM code list as *skos:ConceptScheme* and *skos:Collection*

Each code list contained within LADM is represented by this metamodel as a *skos:ConceptScheme* where this scheme contains a list of all code list values associated with the code list itself. Each code list is in addition defined as a member of a collection through *skos:member*. Each code list has a name, represented by a *skos:prefLabel*, a unique and persistent identifier, represented by a *skos:identifier* and includes a description as defined by the LADM standard. Code list values are of *rdf:type* and *skos:Concept*; each contained within the concept scheme is related to the code list through *skos:inScheme*.

In order to capture the versioning of the code lists over time, a version number for the code list is represented by *dct:hasVersion* and also includes a *prov:created* date to represent the time at which the list becomes available for use within the context of the LADM standard and *dct:modified* to allow for the tracking of changes and/or updates made to the code list over time. With each modification, a new version of the code list is instantiated. Should it be required, it is also possible to represent the invalidation of the code list through *prov:invalidatedAtTime* value.

One of the requirements for the development of this metamodel was the ability to implement code lists according to country specific implementations. In order to capture the country or organisation to which a specific implementation of a code list can be attributed to, the metamodel for the code list includes an attribute *prov:wasAttributedTo*. This attribution should be to a *prov:Agent*.

With the metamodel presented here, it is also possible to capture the fact that a code list can be made up on a collection of code list values by associating a code list to a predefined collection of code list values. These code list values are attributed to a specific *skos:Collection* by defining each a *skos:member* and relating this collection to a defined code list. In this way, a code list can be both a *skos:ConceptScheme* and a *skos:Collection*. Note that the implementation of each is slightly different in practice. It is also possible to capture the fact that every LADM code list can itself be a *skos:member* of a broader collection of code lists. The inclusion of this detail in the metamodel allows for (a given number of) code lists within the LADM to be aggregated into a single collection and this collection can be related to other collections of code lists that may exist, for example, within a different (external) standard, vocabulary or implementation.

Table 2. LADM code list: LA_MortgageType (ISO 19152, 2012)

```
prefix con: <http://www.isoladm.org/model/con/>
prefix dct: <http://purl.org/dc/terms/>
prefix ladm: <http://www.isoladm.org/>
prefix prov: <http://www.w3.org/ns/prov#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix scheme: <http://www.isoladm.org/model/scheme/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

##prefixes for external models should be defined, stdm is just an example:
prefix stdm: <https://stdm.gltn.net/>

## Code List - LADM Mortgage Type

scheme:LA_MortgageType
  a skos:ConceptScheme;
  dct:isFormatOf ladm:Mortgage;
  skos:prefLabel "mortgage types"@en;
  skos:member ladm:LA_CodeListCollection_Administrative;
  rdfs:isDefinedBy "http://www.examplesource.com/mortgagetypes"^^xsd:anyURI;
  skos:relatedMatch stdm:Collateral;
  skos:identifier "http://www.isoladm.org/model/scheme/la_mortgagetype"^^xsd:anyURI;
  dct:description "The LA_MortgageType code list includes all the various mortgage types,
such as linear or micro-credit, used in a specific land administration profile
implementation. The LA_MortgageType code list is required only if the attribute type in
LA_Mortgage class is implemented. Code list shall provide a complete list of all codes with a
name and description"@en;
  dct:hasVersion "1";
  prov:created "01-01-2022"^^xsd:date.
```

In the example presented in Figure 2, the code list has not been modified or invalidated since its creation and, therefore, only has a single version and a creation date. In general, modifications to the code list happen at the level of the code list values and their associated definitions, the details of which are discussed in the following subsection. The only alternative to this rule is when an entire code list is invalidated by a change to the LADM standard and new and/or alternative code list values are included, the consequences of which are outlined in the section 3.1.4.

3.1.2 LADM code list value as *skos:Concept*

Each code list value associated with a given code list in the LADM is represented according to this metamodel as a *skos:Concept* with a unique *skos:identifier*. Each code list value is related to its associated code list through the relation *skos:inScheme*. For each code list value, in a similar manner to the code list concept scheme to which the value is associated, a *skos:prefLabel* as well as a *dct:created* date and *dct:modified* date. In a similar manner to the code list itself, the *prov:wasAttributedTo* property allows for a record of which actor specifically defined and implemented this code list value with this particular definition (see Table 3).

One of the main contributions of this metamodel is the explicit inclusion of a *skos:definition* relation to the code list value where a definition (*skos:definition*), a source of the definition (*rdfs:isDefinedBy*) and versioning of the code list (*dct:hasVersion*) can be implemented. Each change to a definition or attribute associated with a code list changes the version number of the code list value and updates the *dct:modified* value for the code list value. This inclusion supports the tracing of definitions of code list values and supports users of the definition with assessing whether the definition is appropriate for implementation in a given context. This change also updates the *dct:modified* on the code list itself. Changes which result in a new version of the code list leave the unique identifier for the code list value unchanged. If significant changes to the code list value needs to be done, a new identifier for the code list value should be applied.

Each code list value allows for the definition of an attributed source through the attribute *prov:wasAttributedTo*. This attribute is intended to be implemented completely independently of the *rdfs:isDefinedBy* attribute where, in a similar manner to the same attribute at the code list level, this attribute explicitly defines which country or organisation originally developed this value definition. This attribute would, therefore, highlight which country (or organisation) has developed this value and which source they used its development. This attribute would, however, not necessarily be equal to the same attribute at the code list level in order to capture the ability for a given country or organisation to make use of existing definitions originally implemented by another country or organisation.

Table 3. Code list values for LA_MortgageType (ISO 19152, 2012)

```
prefix con: <http://www.isoladm.org/model/con/>
prefix country: <http://www.isoladm.org/model/country/>
prefix dct: <http://purl.org/dc/terms/>
prefix ladm: <http://www.isoladm.org/>
prefix prov: <http://www.w3.org/ns/prov#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix scheme: <http://www.isoladm.org/model/scheme/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

## LADM 'Core' Code List Values for LA_MortgageType
con:LA_linear
  a skos:Concept;
  skos:inScheme scheme:LA_MortgageType;
  skos:identifier "http://www.isoladm.org/model/con/la_linear"^^xsd:anyURI;
  prov:wasAttributedTo country:LA;
  skos:prefLabel "linear"@en;
  dct:created "01-01-2022"^^xsd:date;
  dct:hasVersion "1";
  skos:definition "The type of  mortgage with the gradually decreasing periodic payment"@en;
  rdfs:isDefinedBy "http://www.examplesource.com/mortgagedefinition/linear"^^xsd:anyURI .

con:LA_levelPayment
  a skos:Concept;
  skos:inScheme scheme:LA_MortgageType;
  skos:identifier "http://www.isoladm.org/model/con/la_levelPayment"^^xsd:anyURI;
  prov:wasAttributedTo country:LA;
  skos:prefLabel "level payment"@en;
  dct:created "01-01-2022"^^xsd:date;
  dct:hasVersion "1";
  skos:definition "A level payment mortgage is a type of mortgage that requires the same
payment each month or payment period"@en;
  rdfs:isDefinedBy "http://www.examplesource.com/mortgagedefinition/levelpayment"^^xsd:anyURI
.

con:LA_microcredit
  a skos:Concept;
  skos:inScheme scheme:LA_MortgageType;
  skos:identifier "http://www.isoladm.org/model/con/la_microcredit"^^xsd:anyURI;
  prov:wasAttributedTo country:LA;
  skos:prefLabel "microcredit"@en;
  dct:created "01-01-2022"^^xsd:date;
  dct:hasVersion "1";
  skos:definition "Microcredit, a means of extending credit, usually in the form of small 
loans with no collateral"@en;
  rdfs:isDefinedBy "http://www.examplesource.com/mortgagedefinition/microcredit"^^xsd:anyURI.
```

In the example presented above, the hierarchy of concepts contained within a code list has not been explicitly defined. In the metamodel presented, this hierarchy between concepts within a code list can be applied. The following example, which implements a country-specific code list for the LA_AdministrativeSourceType, highlights how this can be done through the implementation of *skos:narrower*, *skos:broader* and *skos:related* associations between concepts where an external code list value has been applied to an existing code list. These relations can be done at the code list level (Table 4) and the code list value level (Table 5).

The inclusion of this possibility within the context of the LADM is important where country-specific implementations have been done and where these implementations are more specific versions of existing LADM code list values. These country-specific implementations are still required to have value definitions and can be versioned in much the same way as the core code list values. A more detailed overview of this process is discussed in the sections that follow. All code lists can themselves be a part of (*skos:member*) of a code list collection. In Table 4, this aggregation is shown were the LA_AdministrativeSourceType code list is a *skos:member* of the code list collection LA_CodeListCollection_Administrative.

Table 4. Application of *skos:related* country specific code list to LADM code list

```
prefix col: <http://www.isoladm.org/model/col/>
prefix con: <http://www.isoladm.org/model/con/>
prefix country: <http://www.isoladm.org/model/country/>
prefix dct: <http://purl.org/dc/terms/>
prefix scheme: <http://www.isoladm.org/model/scheme/>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix ladm: <http://www.isoladm.org/>
prefix prov: <http://www.w3.org/ns/prov#>

scheme:LA_AdministrativeSourceType skos:related scheme:MG_AdministrativeSourceType

scheme:LA_AdministrativeSourceType
  a skos:ConceptScheme;
  dct:isFormatOf ladm:AdministrativeType;
  skos:prefLabel "responsibility types"@en;
  skos:member col:LA_CodeListCollection_Administrative;
  prov:wasAttributedTo country:LA;
  rdfs:isDefinedBy "http://www.examplesource.com/mongolia_administrativetypes"^^xsd:anyURI.

scheme:MG_AdministrativeSourceType
  a skos:ConceptScheme;
  dct:isFormatOf ladm:AdministrativeType;
  skos:prefLabel "responsibility types for mongolian implementation"@en;
  skos:member col_mg:MG_CodeListCollection_Administrative;
  prov:wasAttributedTo country:MG;
  rdfs:isDefinedBy "http://www.examplesource.com/mongolia_administrativetypes"^^xsd:anyURI.
```

In the example above (Table 4), the use of the relation *skos:narrower* allows for a generation of country-specific implementation of the LA_AdministrativeSourceType code list, which in this case is a more specific implementation of the code list by Mongolia by adding two other code list values to the existing code list to make it more specific in order to suit the country-specific context. The details of this implementation are shown in Table 5. Please note the use of the country code MG_ to denote a country-specific implementation of a code list and the aggregation of the code list to a country-specific collection through *skos:member* relations.

Please note, in Table 5, definitions for the code list values are not defined. This is because definitions for these values were not explicitly defined in the source paper and so they were not reproduced for the purpose of this paper in order to remain in complete alignment with the source for demonstration purposes. Should a definition be available, it should be included in line with the proposed metamodel.

Table 5. Application of *skos:related*, *skos:broader* and *skos:narrower* attributes to the country-specific code list MG_AdministrativeSourceType based on Munkhbaatar et al. (2018)

```
prefix col: <http://www.isoladm.org/model/col/>
prefix con: <http://www.isoladm.org/model/con/>
prefix def: <http://www.isoladm.org/model/def/>
prefix ladm: <http://www.isoladm.org/>
prefix prov: <http://www.w3.org/ns/prov#>
prefix scheme: <http://www.isoladm.org/model/scheme/>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

scheme:MG_AdministrativeSourceType
  a skos:ConceptScheme;
  skos:prefLabel "administrative source types for mongolian implementation"@en;
  skos:member col:MG_CodeListCollection_Administrative;
  prov:wasAttributedTo country:MG;
  rdfs:isDefinedBy "http://www.examplesource.com/mongolia_administrativetypes"^^xsd:anyURI.

## Code List Values for MG_AdministrativeSourceType (Country-Specific Implementation)

con:MG_morrtgage # as implemented in source paper
  a skos:Concept;
  skos:inScheme scheme:MG_AdministrativeSourceType;
  prov:wasAttributedTo country:MG;
  rdfs:isDefinedBy
"http://www.examplesource.com/mongolia_administrativetypes/morrtgages"^^xsd:anyURI;
  skos:identifier "http://www.isoladm.org/model/con/mg_morrtgage"^^xsd:anyURI;
  skos:related con:LA_mortgage;
  skos:prefLabel "morrtgage"@en;
  dct:created "01-01-2022"^^xsd:date .

con:MG_landCertificate
  a skos:Concept;
  skos:inScheme scheme:MG_AdministrativeSourceType;
  prov:wasAttributedTo country:MG;
  rdfs:isDefinedBy
"http://www.examplesource.com/mongolia_administrativetypes/morrtgages"^^xsd:anyURI;
  skos:identifier "http://www.isoladm.org/model/con/mg_landcertificate"^^xsd:anyURI;
  rdfs:isDefinedBy
"http://www.examplesource.com/mongolia_administrativetypes/landCertificate"^^xsd:anyURI;
  skos:prefLabel "land certificate"@en;
  skos:narrower con:LA_deed;
  dct:created "01-01-2022"^^xsd:date .

con:MG_titleDeed
  a skos:Concept;
  skos:inScheme scheme:MG_AdministrativeSourceType;
  prov:wasAttributedTo country:MG;
  rdfs:isDefinedBy
"http://www.examplesource.com/mongolia_administrativetypes/titleDeed"^^xsd:anyURI
  skos:identifier "http://www.isoladm.org/model/con/mg_titledeed"^^xsd:anyURI;
  skos:prefLabel "title deed"@en;
  skos:related con:LA_title;
  skos:narrower con:deed;
  skos:broader con:MG_landCertificate;
  dct:created "01-01-2022"^^xsd:date .

con:MG_landPermissionDecision
  a skos:Concept;
  skos:inScheme scheme:MG_AdministrativeSourceType;
  prov:wasAttributedTo country:MG;
  rdfs:isDefinedBy
"http://www.examplesource.com/mongolia_administrativetypes/landPermissionDeed"^^xsd:anyURI
  skos:identifier "http://www.isoladm.org/model/con/mg_landpermissiondecision"^^xsd:anyURI;
  skos:prefLabel "land permission decision"@en;
  skos:narrower con:LA_agriConsent;
  dct:created "01-01-2022"^^xsd:date .

con:MG_landRegisterAttachedDoc
  a skos:Concept;
  skos:inScheme scheme:MG_AdministrativeSourceType;
  prov:wasAttributedTo country:MG;
  rdfs:isDefinedBy
"http://www.examplesource.com/mongolia_administrativetypes/landRegisterAttachedDoc"^^xsd:anyURI
  skos:identifier "http://www.isoladm.org/model/con/mg_landregisterattacheddoc"^^xsd:anyURI;
  skos:prefLabel "land register attached document"@en;
  skos:narrower con:LA_agriNotaryStatement;
  dct:created "01-01-2022"^^xsd:date .
```

3.1.4 Versioning using the LADM code list metamodel - *skos:identifier*

Although versioning in the context of the proposed metamodel for LADM code lists has been discussed in previous subsections, it is the intention of this subsection to provide a more detailed overview of the way in which versioning is implemented across an entire code list. There are a number of central elements to the versioning of code lists and element is briefly discussed for their role in the next subsections.

Each code list, code list value and value definition has an associated persistent identifier documented based on the attribute *skos:identifier*. In practice any significant changes to a concept, such as a change to a definition or an update to the primary source, would result in a new version of the concept and, therefore, a new identifier. An example of how this might work in practice for a given code list value and associated definition can be found in the figure below (Table 6). A comprehensive IRI strategy is beyond the scope of this paper but should be defined and applied in future iterations of this work.

The inclusion of this attribute is important for the management of versioning within code lists because the persistence identifier ensures that versioning of the concept does not necessarily affect the concept itself. Indeed, any significant change to the concept would result in a new identifier for the concept; supporting the confidence in using and reusing the code lists in a given implementation across time. Separating value definitions from their associated value and providing each with a persistent identifier also allows for flexibility in the implementation of definition across various contexts while still supporting traceability of these concepts and relations across time and across context.

Table 6. LADM coe list values for LA_MortgageType with versioning examples (ISO 19152, 2012)

```
prefix con: <http://www.isoladm.org/model/con/>
prefix country: <http://www.isoladm.org/model/country/>
prefix dct: <http://purl.org/dc/terms/>
prefix ladm: <http://www.isoladm.org/>
prefix prov: <http://www.w3.org/ns/prov#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix scheme: <http://www.isoladm.org/model/scheme/>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

## Value Definitions for Code List Values associated with LA_MortgageType

con:LA_linear
  a skos:Concept;
  skos:identifier "http://www.isoladm.org/model/con/la_linear/1"^^xsd:anyURI;
  skos:definition "The type of  mortgage with the gradually decreasing periodic payment"@en;
  prov:wasAttributedTo country:LA;
  rdfs:isDefinedBy "http://www.examplesource.com/mortgage/linear"^^xsd:anyURI;
  dct:hasVersion "1";
  dct:created "01-01-2022"^^xsd:date.

con:LA_linear
  a skos:Concept;
  skos:identifier "http://www.isoladm.org/model/con/la_linear/2"^^xsd:anyURI;
  skos:definition "The type of  mortgage with the gradually decreasing periodic payment over
the course of defined period of time"@en;
  prov:wasAttributedTo country:LA;
  rdfs:isDefinedBy "http://www.examplesource.com/mortgage/linear-2"^^xsd:anyURI;
  dct:hasVersion "2";
  dct:created "01-02-2022"^^xsd:date.
```

3.1.5 *dct:hasVersion*

The way in which *dct:hasVersion* is implemented across the model only differs in the rules which govern the generation of a new version of a given code list or code list value.

At the level of the code list, the *dct:hasVersion* attribute allows for the differentiation between different versions of the same code list. A new version number is generated when there is an addition or removal of a code list value from the code list and a new version number is generated in line with a change to the *dct:modified* date. The core identifier of a given code list remains stable and the identifier for different versions of the same code list is generated based on the combination of the identifier and version number. An example of this identifier strategy is illustrated above (Table 6).

At the level of code list value, *dct:hasVersion* allows for the differentiation between different versions of the same code list value based on differences in the definition associated with the code list value. Minor changes to the definition of a code list value will only change the version of the code list value. Changes to the original source of the definition, a fundamental change to the meaning of the definition used or a change to the country or organisation to which a given code list value with a given definition is associated will result in a new code list value and a new identifier for that value. In this way, a code list value may have various definitions, all with their own versions, associated with different organisations and/or countries in different contexts.

3.1.6 Versioning dates with *dct:created* and *dct:modified*

In a similar manner to the identifiers, each code list and code list value has an associated *dct:created* date based on the date on which the given entity was first instantiated. At the level of the code list, the *dct:modified* attribute tracks changes to the code list itself based on the addition or removal of a code list value from that list. As such, the last modified date is the same as the latest *dct:created*/*dct:modified* date across all code list values in a code list. At the level of code list value, updates *dct:modified* only happen when a code list value is being versioned. There are no other influences on this date.

3.1.7 *prov:invalidatedAtTime*

The attribute *prov:invalidatedAtTime* is used at all levels of the metamodel to signal the end of the validity of a particular code list or code list value. The way in which this attribute is implemented at a given level influences the implementation of this attribute at other levels.

At the code list level, the *prov:invalidatedAtTime* attribute denotes the point at which the code list itself is no longer valid for implementation. When this time is actually instantiated is dependent on when the standard itself was changed and a code list was invalidated or when a country-specific implementation of a code list was invalidated by that country or organisation. The instantiation of this attribute at the code list level does not necessarily invalidate the code list values contained within the code list. As such, different versions of a code list can have the same version of a code list value.

At the level of code list value, the *prov:invalidatedAtTime* attribute is used when either the standard or a country-specific implementation explicitly invalidates a code list value. Additionally, making use of the *prov:invalidatedAtTime* attribute at the level of code list value is intended to be used when a new version of a definition explicitly replaces an older version. Because of the variety of implementations across various contexts, it is the intention that this attribute is not frequently used and that versions of definitions remain valid for implementation across time. This allows for flexibility in implementation and easier maintenance of implemented code list values and their definitions across time.

3.1.8 Example implementation of the metamodel

In order to provide some insight into a small implementation of this approach in practice, the code list for LA_MortgageType as defined above was uploaded to a triple store, in this case, a TriplyDB instance, and the code list and its values made available in a SPARQL endpoint. This endpoint [12] then queried using the SPARQL query defined below to produce the result as illustrated in the Table 7. The semantic information and structure might be overwhelming to users of the model, so it is also investigated how to extract a simple set of values for the code lists. If more information is needed, then the user can explore the semantic information behind a simple value.

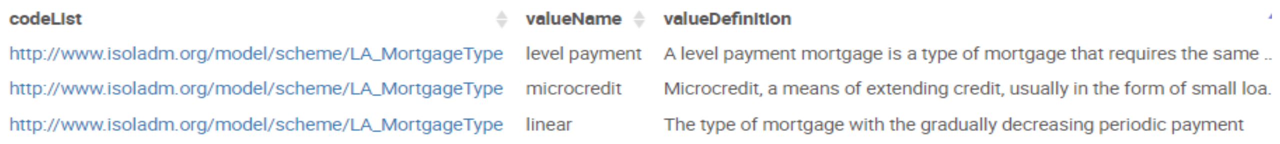
[12] https://data.labs.kadaster.nl/experiment/-/queries/Query/3

Table 7. SPARQL query for retrieving a list of code list values

```
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select ?codeList ?valueName ?valueDefinition where {
  ?codeList
    a skos:ConceptScheme;
          skos:prefLabel ?codeListName .
  ?codeListValue
    a skos:Concept;
          skos:inScheme ?codeList;
          skos:prefLabel ?valueName;
    skos:definition ?valueDefinition .
}
limit 10
```

Figure 2 presents the returned results from the query that listed the values of LA_MortgageType code list.



codeList	valueName	valueDefinition
http://www.isoladm.org/model/scheme/LA_MortgageType	level payment	A level payment mortgage is a type of mortgage that requires the same ..
http://www.isoladm.org/model/scheme/LA_MortgageType	microcredit	Microcredit, a means of extending credit, usually in the form of small loa
http://www.isoladm.org/model/scheme/LA_MortgageType	linear	The type of mortgage with the gradually decreasing periodic payment

Figure 2. The retrieved result of the query

4. CONCLUSION

Code lists can be considered as a supplement to the classes and associations within a standard like ISO 19152 LADM. They present a simple list of example values without any structure, definition, reference to the source of a definition, multi-lingual alternative term and semantic relationships (e.g. hierarchical).

This paper investigates the requirements for structuring and managing semantically enriched code list and their values. The identified requirements for refined code list values are basically derived from ISO 19150-2:2015 standard and experts' opinions. The relevant thesauri and registries are identified, and their solutions are evaluated against the identified requirements. Furthermore, this paper proposes a metamodel using the identified requirements. It is considered that the proposed metamodel may be good for starting a discussion about the management and maintenance of refined code list at the conference. An agreed version of code list metamodel may be included in the new edition of LADM.

Testing the metamodel with more complete use cases (e.g. all LADM code lists, code list values and their definitions, and a more complete country profile) is planned as a future work activity. This activity will be initiated after the metamodel structure and content are agreed upon at the conference. Moreover, a more strict methodology to extend metamodel are going to be specified including IRI strategy for country profile implementation. Furthermore, it should also be decided which registries can be used to publish and maintain code list values of LADM, considering the preferences of existing thesauri. In addition, publishing all LADM code lists and their values as SPARQL endpoint can be seen as a reasonable future work since it enables querying code list values on the Web and increases the discoverability of published code lists values that facilitate the interoperability across domains.

REFERENCES

Çağdaş V., Meggiolaro L. & Stubkjær, E. (2021). Semantic Resources for the Geospatial Domain. FIG e-Working Week 2021, Smart Surveyors for Land and Water Management - Challenges in a New Reality, Virtually in the Netherlands, 21–25 June 2021.

Çağdaş, V., & Stubkjær, E. (2015). A SKOS vocabulary for linked land administration: cadastre and land administration thesaurus. Land Use Policy, 49, 668-679.

Ciparisse, G., 2003. Multilingual thesaurus on land tenure. FAO. ISBN: 9251042837. Available at: http://www.fao.org/publications/card/en/c/ea6f2295-e117-5ac0-9a85-caa16cd50fd5/

ISO 19150-2:2015: Geographic information — Ontology — Part 2. https://www.iso.org/standard/57466.html

ISO 28258:2013. Soil quality — Digital exchange of soil-related data. https://www.iso.org/standard/44595.html.

Munkhbaatar, B., Kim, M. G., Lee, Y. H., & Koh, J. H. (2018). A Study on the Design of LADM-based Cadastral Data Model for Mongolia. Journal of Cadastre & Land InformatiX, 48(2), 51-64.

Paasch, J. M., van Oosterom, P., Lemmen, C., & Paulsson, J. (2015). Further modelling of LADM's rights, restrictions and responsibilities (RRRs). Land Use Policy, 49, 680-689.

Stubkjær, E., & Çağdaş, V. (2021). Alignment of standards through semantic tools–The case of land administration. Land Use Policy, 104, 105381.

Stubkjær, E., Gruler, H. C., Simmons, S., & Cagdas, V. (2019). Code list management supported through a controlled domain vocabulary. In Proceedings of the 8th International FIG Workshop on the Land Administration Domain Model.

Stubkjær, E., Paasch, J. M., Çağdaş, V., van Oosterom, P. J. M., Simmons, S., Paulsson, J., & Lemmen, C. H. J. (2018). International code list management–the case of land administration. In Proceedings of the 7th International FIG Workshop on the Land Administration Domain Model.

van Oosterom, P., Lemmen, C., Thompson, R., Janečka, K., Zlatanova, S. ve Kalantari, M. (2018). 3D cadastral information modelling. van Oosterom, P. (Ed) FIG Best Practices 3D Cadastres, 95-133. Copenhagen, Denmark.

BIOGRAPHICAL NOTES

Abdullah Kara holds BSc in Geomatics Engineering from Istanbul Technical University and MSc degree in Geomatics Programme of Yıldız Technical University (YTU). He worked as an engineer in the Development of Geographical Data Standards for Turkey National GIS Infrastructure. He received a PhD from YTU in 2021. During his PhD, he visited GIS Technology Section, Department OTB, Delft University of Technology as a guest researcher in 2018. Currently, he is a postdoctoral researcher at Delft University of Technology.

Alexandra Rowland obtained an MSc in Geographical Information Management and Applications from the University of Utrecht, the Netherlands in 2021 based on the thesis proposing a methodology for automatic GIS workflow generation using semantic technologies. Following graduation, Alexandra joined Kadaster, the Dutch Land Registry and Mapping Agency, as a PDEng student and is currently supporting the ongoing development of the Kadaster Knowledge Graph. Her research investigates the role that this Knowledge Graph and other linked data developments within the organisation could play in Kadaster's future data platform.

Peter van Oosterom obtained an MSc in Technical Computer Science in 1985 from Delft University of Technology, the Netherlands. In 1990 he received a PhD from Leiden University. He is professor at the Delft University of Technology, and head of the 'GIS Technology' Section, Department OTB, Faculty of Architecture and the Built Environment, Delft University of Technology, the Netherlands. He is the current chair of the FIG Working Group on '3D Cadastres'.

Erik Stubkjær is emeritus professor, having served as professor of cadastre and land law at Department of Development and Planning, Aalborg University 1977 – 2008. He is engaged in standardization activities, contributing to the OGC standards LandInfra and InfraGML (2016/17). He graduated as land surveyor in 1964 and obtained his Ph.D in 1969. During 1979-1988, he was a member of the Tribunal of the Danish Association of Chartered Surveyors.

Volkan Çağdaş is full professor for cadastre and land administration at Yildiz Technical University, Department of Geomatic Engineering, Istanbul / Turkey. He has been teaching cadastre, immovable property law, land re-adjustment, real estate valuation, and land information management systems at undergraduate and graduate levels. His research interest focuses on the technical and the institutional aspects of cadastre and land administration.

Erwin Folmer received his MSc in Technical Business Administration (Industrial Engineering) in 1999 at the University of Twente, based on a master thesis assignment on requirements engineering at Baan Development. From 1999 until 2001 Erwin was innovator at KPN Research involved in amongst other the order entry and billing systems of ADSL services. In 2001 Erwin joined TNO, and became senior scientist on the topic of interoperability and standards. From 2009 he part-time joined the University of Twente to start a PhD research on the standardization topic, while continuing his work for TNO. In 2012 received his PhD based on the 'Quality of Semantic Standards' thesis. In 2013-2014 Erwin was visiting researcher at ERCIS/University of Munster. From 2015 onwards Erwin joined Kadaster. At Kadaster he is leading the developments of the Linked Data platform for open geographical data. Currently this is the largest deployment of Linked Data in the Netherlands, and among the largest in the world.

Christiaan Lemmen is full Professor Land Information Modeling at the Faculty of GeoInformation Science and Earth Observation of the University of Twente in the Netherlands. His other main job is as Senior Geodetic Advisor at Kadaster International, the international branch of the Netherlands Cadastre, Land Registry and Mapping Agency. He is director of the OICRF, the International Office of Cadastre and Land Records, one of the permanent institutions of the International Federation of Surveyors (FIG).

Wilko Quak has an MSc in computer science from Utrecht University, The Netherlands (UU). He worked for several years (1993-2001) as a researcher at the Dutch research center for mathematics and computer science (CWI) and University of Amsterdam (UvA) on Spatial DBMS performance. Since 2001 he has been a researcher at the Section GIS Technology, Delft University of Technology. At Delft University his research focus is moving towards spatial data modeling, data interoperability and standardization. Since 2007 he has been working part-time for Geonovum (the National Spatial Data Infrastructure (NSDI) executive committee in the Netherlands).

Laura Meggiolaro is Team Leader at the Land Portal Foundation. Over the past 16 years Laura has been working mainly for the land governance sector specialising in information and data management for development with an increasing interest in Open Data, semantic technologies and open standards. She holds a master in communication science and master in economics for developments. Since 2011 Laura has been responsible for the overall management, implementation and expansion of the Land Portal contributing to the process that has seen the Land Portal evolving from a project into the independent organization that maintains the Land Portal website: the single one-stop-shop about land governance data and information and a vibrant convening and knowledge exchange platform that promotes transparency and modern open data ecosystems. Prior to coming to the Land Portal Foundation, she has been working with the Food and Agriculture Organization, ILC at the International Fund for Agricultural Development and ActionAid International, specialising in information and knowledge management for land rights.

CONTACTS

Abdullah Kara
Delft University of Technology
Faculty of Architecture and the Built Environment
P.O. Box 5030
2600 GA Delft
THE NETHERLANDS
E-mail: A.Kara@tudelft.nl

Alexandra Rowland
Cadastre, Land Registry and Mapping Agency Kadaster International
P.O. Box 9046
7300 GH Apeldoorn
THE NETHERLANDS
E-mail: Erwin.Folmer@kadaster.nl
Website: www.kadaster.nl

Peter van Oosterom
Delft University of Technology
Faculty of Architecture and the Built Environment
P.O. Box 5030
2600 GA Delft
THE NETHERLANDS
E-mail: P.J.M.vanOosterom@tudelft.nl

Erik Stubkjær
Aalborg University
Department of Planning
Rendsburggade 14, niveau 3
DK-9000 Aalborg,
DENMARK
Phone: +45 23319553
E-mail: est@plan.aau.dk
Website: https://vbn.aau.dk/en/persons/erik-stubkjaer

Volkan Çağdaş
Yıldız Technical University
Department of Geomatic Engineering,
34210 Istanbul
TURKEY
Phone: + 90 212 383 5313
Fax: + 90 212 383 5274
E-mail: volkan@yildiz.edu.tr
Website: https://avesis.yildiz.edu.tr/volkan

Erwin Folmer
Cadastre, Land Registry and Mapping Agency Kadaster International
P.O. Box 9046
7300 GH Apeldoorn
THE NETHERLANDS
E-mail: Erwin.Folmer@kadaster.nl
Website: www.kadaster.nl

Christiaan Lemmen
University of Twente
Faculty of Geo-Information Science and Earth Observation/ITC
P.O. Box 217
7500 AE Enschede
THE NETHERLANDS
Phone: + 31 6 52481717
E-mail: c.h.j.lemmen@utwente.nl
Website: www.itc.nl
And
Cadastre, Land Registry and Mapping Agency Kadaster International
P.O. Box 9046
7300 GH Apeldoorn
THE NETHERLANDS
Phone: +31 88 183 4417
E-mail: Chrit.Lemmen@kadaster.nl
Website: www.kadaster.nl

Wilko Quak
Geonovum
Barchman Wuytierslaan 10
3818 LH Amersfoort
THE NETHERLANDS
E-mail: C.W.Quak@tudelft.nl

Laura Meggiolaro
Land Portal Foundation
Via Fonteiana
Rome
ITALY
Phone: +00393471246461
E-mail: laura.meggiolaro@landportal.info
Website: https://landportal.org/user/41