Guidelines for translations of IFLA namespaces in RDF

IFLA Namespaces Technical Group

**Version 2.0, August 2013**

# Introduction

These guidelines are intended to support the translations of the namespaces containing representations of IFLA standards in Resource Description Framework (RDF) from their original language into multiple other languages.

Translation is appropriate for the textual strings, mainly untyped literals, forming the object part of the triples used to describe element sets of RDF classes and properties and value vocabularies of concepts represented in Simple Knowledge Organization System (SKOS). These include:

* Human-readable labels, names, titles, etc.
* Definitions, descriptions, etc.
* Notes, comments, etc.

# General guidelines

The content of labels, definitions, notes, and other literals in a namespace should match the text of the standard in its original language as closely as possible, but adapted as necessary to make the content understandable in the context of RDF labels, etc. The main benefit of matching the text in the namespace and textual documentation is to clarify their relationship and allow the documentation to be searched for additional information using the terminology of the namespace.

Translations of namespace content should be as close as possible to the source wording in the namespace, without losing coherency of meaning in the translated language and retaining consistency with the corresponding official translation of the source documentation of the standard.

It is recommended that the Multilingual Dictionary of Cataloguing (MulDiCat) is used as a source of terminology for IFLA bibliographic namespaces and their translations. If a suitable term is missing from MulDiCat, it should be proposed for inclusion.

# Specific guidelines

## Scope

It is not necessary to translate the entire namespace in one go. Translation of any part will immediately benefit users. Version and status control should be used to inform users and manage a multi-phase approach to translation. Factors affecting the priority assessment of common RDF literals suitable for translation are given in the following table.

|  |  |  |
| --- | --- | --- |
| Literal element | Primary user | Usage |
| rdfs:comment | Application technical/service developer | Contains the primary semantic of the element, usage and scope notes, etc.  |
| rdfs:label | Application technical developer | Aide-memoire for machine-readable URI opaque to human readability. |
| reg:name | Application technical developer | Aide-memoire for URI, with no embedded spaces. The reg:name property is specific to the IFLA namespaces registry and is part of its namespace. It is usually a camelCase version of rdfs:label. |
| skos:altLabel | End-user | Contains an alternative, non-preferred version of a preferred label or authorized “heading”. |
| skos:definition | Application technical/service developer; end-user | Contains the primary semantic of the element, necessary for determining the correct element to use in an application, and useful for end-user help functions. |
| skos:prefLabel | End-user | Contains the authorized “heading” or other preferred label, name, title, etc. |
| skos:scopeNote | Application technical/service developer; end-user | Contains additional semantics of the element, important for determining correct usage, and useful for end-user help functions. |

The priorities for developers and end-users are different, as are priorities for each type of vocabulary.

Generally, the priority for element sets is definition and label; these are important for the development of applications using the namespace. The priority for value vocabularies is preferred and alternative label and definition; these are important for end-users. However, the full benefit to a multilingual Semantic Web is realized only when all elements and concepts are translated.

In the current development phase of the Semantic Web, translation of element set labels offers a high cost-benefit by encouraging participation in, and attention to, the multilingual environment.

## Style

Use upper, lower, and camelCase following Semantic Web conventions, but take into account language and cultural norms. The unofficial convention is to use lower-case initial letters for all words except proper nouns in property labels, and to use upper-case initial letters for all words in class labels. Adjustments should be made if the result is unacceptable in the context of the translated language. For example, the ISBD namespace includes classes and properties for aggregated statements; that is compounds of lower level elements. The English labels for these classes are constructed using a pattern of element label followed by “encoding scheme”; the element label acts as an adjectival phrase. This results in lengthy labels where every word is capitalized, following the convention for class labels. When camelCase is applied to obtain the OMR name, the result can be very ugly in translation. For example “ParallelTitleCompoundEncodingScheme” is quite readable in English, which is very tolerant of the use of noun-adjectives, but is unacceptable for direct translation into Spanish.

CamelCase or Camel case is the practice of writing phrases in which the spaces between words are removed and the initial letter of each word is capitalized (the first letter of the first word may be in either upper or lower case); e.g. “thisIsCamelCase”. This avoids embedded spaces in phrases which may form part of a URI that can be processed as a URL in a standard web browser for de-referencing purposes; spaces in URLs must be encoded as “%20”, so removing them is intended to keep the phrase readable and has the benefit of shortening the URI/URL. CamelCase is only recommended for labels intended to be used for shorthand and technical indexing purposes.

Use verbal phrases for general property labels. The purpose is to reflect the use of a property as a predicate in an RDF data triple; the subject-predicate-object construction accommodates single, basic statements in the form of subject-verb-object. A verbal phrase can be derived from a noun phrase used as an attribute label in the source documentation by prefixing it with “has …” or bracketing it with “is … of”, or similar consistent methods.

Avoid the use of definite and indefinite articles in labels, in order to keep them concise, unless this is unacceptable in the translation language.

Prefer singular over plural forms of words.

## Reference source

There are two separate, but related, sources of the namespace translations:

1. The documentation of the standard schema or terminology, translated from the original language.
2. The RDF namespaces derived from that documentation, in the original language. The namespace content is translated directly.

If source 1 is used, a standard method should be employed to derive the namespace content from the translated documentation, following the method used to derive the original namespace from the original documentation.

The use of source 2 assumes that the text in the namespace is unambiguous in the original language. Evidence from the development of the original documentation often shows that this is not the case, and that careful definition of terms and artificial phrasing is necessary to achieve consistency.

The preferred source for translations of the namespace is the translated text of the standard’s current published documentation. For example, the Spanish translation of ISBD first used the English text of the namespace, rather than the official Spanish translation of the ISBD documentation. This presented a number of problems in making the translations reasonably understandable in Spanish. There were also inconsistencies with the documentation translation. For example, it was necessary to add prepositions to labels to make them readable. As a result, it was decided that Spanish developers would prefer to use the official Spanish translation of the standard as a reference source and therefore the namespace should be aligned to the official text in Spanish.

If no such translated text is available, then the namespace literals should be translated directly. It is obviously important that the namespace refers to the current version of the standard. There is often a time-lag between publication and translation, and translations of standard documentation may be significantly out-of-date. This problem is also likely to occur when a standard is being amended; there will be time when the standard, its namespace, and its translations are all out of synchronization, leading to potential incoherency in the namespace.

It is essential to ensure that the semantic coherency of the namespace remains intact. For specific categories of literal:

* Labels should be based directly on the name of the element in the standard documentation. This allows pattern-matching of label to documentation; a label does not carry the main semantic of an element, so semantic coherency is less of an issue.
* Definitions should align with the documentation text but with the necessary modifications to ensure semantic coherency within the namespace.
* The treatment of scope notes can be more flexible. There may be cultural and contextual reasons for significant differences in examples of scope boundaries, as long as the focus of the definition remains coherent.

## Disambiguation

First-order application of a standard method of deriving RDF labels from element names in source documentation may result in identical labels which require disambiguation. For example, in the FRBRer and FRAD namespaces, the label “is part of” is derived from the similarly-named relationship in the FRBR entity-relationship model. But this relationship applies to all four FRBR group 1 entities, requiring four different RDF properties in the FRBR namespace.

A standard method of disambiguation of property labels is to add the names of the classes which are the property’s domain and range. Punctuation and other syntactical devices used may not be suitable in other languages, and translations may need to modify their use. For example, the FR namespaces use brackets; e.g. “is part (item) of”, “is part (work) of”. Where such a property has both a domain and a range using different classes, two bracketed qualifiers are added; e.g. “has a supplement (work) (from expression)”. The use of brackets follows a pattern which exploits the flexibility of English but looks arbitrary and ambiguous in Spanish. After further discussion, acceptable Spanish translations were developed; e.g. “forma parte (ejemplar) de”, “forma parte (obra) de”, and “tiene suplemento (obra) (de expresión)”.

## Value vocabularies

Grammatical inflection is very rare in English, but is a fundamental feature of many other languages. It becomes a problem in a value vocabulary if inflected terms are presented as standalone entries. For example, ISBD has controlled vocabularies for qualifiers of content form in area 0. The qualifiers are given as adjectives to allow them to be used in user-friendly compound phrases; e.g. “image (cartographic)”, where “cartographic” is a qualifier. In Spanish, “cartographic” is inflected in two forms, “cartográfica” and “cartográfico”, that depend on the gender of the noun being qualified: “imagen (cartográfica)” and “objeto (cartográfico)”. The conventional method of representing authoritative terms in SKOS namespaces for value vocabularies is to use the skos:prefLabel property, but only one such property is allowed for each language. The initial ISBD solution was to use an artificial combination of the inflected terms: “cartografico/a”, but this prevented the vocabularies being used directly in user-friendly displays. The current approach used in the ISBD namespace is to record the inflected terms using the skos:altLabel property, which is repeatable for any language. This means there is no skos:prefLabel for the concept in Spanish. The same problem and solution occur with the Croatian translation: “kartografski” and “kartografska” use skos:altLabel. Further monitoring and investigation of better lexicographical solutions is required

Types of term which are inflected in many languages should be avoided as standalone terms in a value vocabulary, if possible.

Where such terms are necessary in a vocabulary, equivalent representation for each language in a namespace may not be possible, and alternative methods will need to be developed.