I18n of an S-100 feature catalogue

January 2022

SevenCs GmbH, Hamburg, Germany

SevenCs GmbH

Atlantic Haus

Zirkusweg 1

D-20359 Hamburg

Tel. +49 (0)40/851 72 40

FAX +49 (0)40/851 72 4-79

http://www.sevencs.com

Introduction

This document discusses the possibilities how elements of a feature catalogue that contains text in a human language can be made available in other languages. This would applications allow such to present such texts in a language chosen by the user. though this document is considering the S-100 feature catalogue, the concepts are universal and can be applied to other resources.

Analysis / Discussion

Internationalisation (often called i18n) are means of adapting computer software to different languages, regional peculiarities, and technical requirements of a target locale. This includes the graphical user interface but also resources which contain text in a human language. Such resources include catalogues and dictionaries.
Without loss of generality this document will discuss the S-100 feature catalogue and methods that can be used to achieve i18n.
Before looking at any technical solutions we consider the processes how feature catalogues for different languages can be created and maintained.
Two fundamental different concepts are:

1. Using one central document that contains all information
2. Using for each language a separate document that provides the necessary information for one language.

Both concepts have advantages and disadvantages.

Central document

The main advantage is that all information is in one resource only and the technical means are relatively simple. A common technique is to provide the language information with any text, like

```
<name lang="eng">Stake, Pole, Perch, Post</name>
<name lang="deu">Stange</name>
```

The main problem of this concept is on the maintenance of such a central document. It is the problem of maintaining one document with many editors. The number of possible combinations becomes huge when many languages should be supported and since each language editor will maintain the document due to corrections the situation becomes even more complicated. Another problem is that the original document is altered continuously and must be made available to the end application after each change. The catalogues volume will also increase significantly by any new language.

Additional Language document

The problems on the maintenance can be much better handled with this approach. Every editor is responsible for only one document, new versions of this document are not in conflict with the changes of other languages documents. The original document will not be altered and had not to resubmitted to the applications. Software vendors can choose the languages which they want to support and only for those the language files must be made available to the software.
It is off course ore complex to synchronize the additional translation files with the versions of the feature catalogue. Nevertheless, this problem seems to be neglectable comparing to the maintenance issues with the centralized approach.
The concept of language packages as addons is widely used in software technology. Technically it is more challenging than the methods used in a central document, but technologies are available and the technical problems are solvable and giving also some kind of flexibility to the concept.

Technical solutions for additional language documents

The following problems must be solved.

1. Identify all elements that are subject of translation
2. Create a translation template for those elements
    a. keep those items identifiable in the main document
    b. merge items in an existing translation file
3. Translate the items, this can be supported by customized tools or done with a simple text editor.
4. Final application must support the translation files and provide an interface to access the translated texts.

One possible solution for the first step is an XSLT files that automatically identifies all elements in the XML feature catalogue and put them in a definition file together with means to identify the element in the original document. For the identification a technology like XPath would be a perfect choice.
Both the XSLT and the schema for the definition file can be provided with the standard. The latter can be used for other resources if the original resources is in XML format.
Note that the definition file must be a physical file it can exists just as a DOM tree in the application.

From this definition file the translation file can be created/merged. In the translation file for each item the following information should be provided.

1. An XPath expression to uniquely identify the original element in the FC
2. The original text.
3. An element for the translated text either empty or already filled in the case of merging
4. A status flag the shows if the element has a valid translation.
   Note that if the original text has changed during the merging the translation status will be changed but the translation will be kept.
5. The translation should be made by means of a tool that lists all items together with its status and allows iterating over all ‘non valid’ items as well as individually editing single items.
6. At the end of the editing process, which can be time consuming, there is a translation file that contains all translated texts each uniquely bound to an element in the original FC.
7. The process of reading and storing this information is application dependent in in the responsibility of the OEM. Note that the translation file itself confirms to a standardized schema. This allows applications to prove (verify) its structure.

Example

Here is a possible structure and content of a translation file.

```
<S100_TranslationFile
  language="deu"
  xmlns:S100FC="http://www.iho.int/S100FC">
  <S100_TranslationItem>
    <path>/S100FC:S100_FC_FeatureCatalogue/S100FC:scope</path>
    <original>Ocean, Coastal, Ports and Harbors. Excludes Inland waters. Supports
QualityOfBathymetricData and 2015 updates to FC model.</original>
    <translation>Falsche Übersetzung</translation>
    <status>changed</status>
  </S100_TranslationItem>
  <S100_TranslationItem>
    <path>/S100FC:S100_FC_FeatureCatalogue/S100FC:fieldOfApplication</path>
    <original>Marine Navigation</original>
    <translation />
    <status>empty</status>
  </S100_TranslationItem>
  <S100_TranslationItem>
  <path>/S100FC:S100_FC_FeatureCatalogue/S100FC:S100_FC_SimpleAttributes/
        S100FC:S100_FC_SimpleAttribute[./S100FC:code/text()='basedOnFixedMarks']/S100FC:name</path>
    <original>Based On Fixed Marks</original>
    <translation />
    <status>empty</status>
  </S100_TranslationItem>

…

  <S100_TranslationItem>
    <path>/S100FC:S100_FC_FeatureCatalogue/S100FC:S100_FC_SimpleAttributes/
          S100FC:S100_FC_SimpleAttribute[./S100FC:code/text()='beaconShape']/S100FC:listedValues/
          S100FC:listedValue[./S100FC:code/text()='1']/S100FC:label</path>
    <original>Stake, Pole, Perch, Post</original>
    <translation>Stange</translation>
    <status>translated</status>
  </S100_TranslationItem>
  <S100_TranslationItem>
    <path>/S100FC:S100_FC_FeatureCatalogue/S100FC:S100_FC_SimpleAttributes/
          S100FC:S100_FC_SimpleAttribute[./S100FC:code/text()='beaconShape']/S100FC:listedValues/
          S100FC:listedValue[./S100FC:code/text()='1']/S100FC:definition</path>
    <original>An elongated wood or metal pole, driven into the ground or seabed, which serves as a
navigational aid or a support for a navigational aid.</original>
    <translation />
    <status>empty</status>
  </S100_TranslationItem>

..

</S100_TranslationFile>
```

Conclusion

The advantages of the translation files are so vast that only this approach should be further investigated. The XSLT for the feature catalogue should be described in Part 5 as an Annex. The general concept needs another place, a section on internationalisation of XML resources that describes the general concept and the schemas for the translation files.