

ITS 2.0 in XLIFF 2

FEISGILTT Dublin June 2014

Yves Savourel

ENLASO Corporation

Why the mapping?

- ITS 2.0 provides many data categories that match or complete XLIFF metadata.
- ITS 2.0 has a mapping to XLIFF 1.2

Having a mapping for XLIFF 2 make sense

- Mapping done by the ITS Interest Group
http://www.w3.org/International/its/wiki/XLIFF_2.0_Mapping
- Goal is to create a new XLIFF 2 module

Types of mapping

- Data categories not used directly in XLIFF (typically non-metadata data categories e.g. Id Value)
- Use existing XLIFF metadata: e.g. Translate
- Use ITS markup directly: e.g. Text Analysis
- Use a mixed mapping: e.g. Terminology

Marker type

- Use `type="its:any"` in most cases
- Mix of data categories can share one annotation of type `its:any`
- Any data category that can use `its:any` can use other marker types too
- Exceptions:
 - Terminology (only `type: term` or `its:term-no`)
 - Localization Note (only `type: comment`)

Translate

- The XLIFF `translate` attribute has the exact same syntax and semantics as in ITS.
- In `<file>`, `<group>`, `<unit>` and `<mrk>/<sm/>` elements.
- [Example](#)

Localization Note

- `<mrk type="comment">` with either a `value` attribute or a `ref` attribute.
- Note that `ref` attribute must point to an internal `<note>` within the unit
- `priority="1"` is `locNoteType="alert"`, other priority values map to `"description"`

Terminology

- `<mrk type="term" ...>` with `ref` mapping to `its:termInfoRef`
- `istxlf:termConfidence` for `its:termConfidence`
- Use `type="its:term-no"` for `its:term="no"`
- Challenging to implement because mix of Core + ITS features (3 different namespaces)
- [Example](#)

Directionality

- **Not mapped yet**
- XLIFF 2.0 has `srcDir`, `trgDir`, `dir` (values: `ltr`, `rtl` or `auto`)
- Inside content: uses Unicode control characters

Language Information

- In `<xliff>` element:
 - Use `xliff@srcLang` for the source language
 - Use `xliff@trgLang` for the target language
- Inline:
 - Use `xml:lang` in `<mrk>`

Elements Within Text

- Not used directly in XLIFF, but it drives what XLIFF element is used when extracting:
- `withinText="no"`: **go to** `<unit>`
- `withinText="yes"`: **go to** `<pc>`, `<sc>`, `<ec>`
or `<ph>`
- `withinText="nested"`: **go to separate** `<unit>`. **With** `subFlows` **attribute in parent.**

Domain

- Use `istxlf:domains` attribute.
- In `<unit>` and `<mrk>` elements
- [Example](#)

Text Analysis

- Use ITS native attributes.
- In `<mrk>` element.
- [Example](#)

Locale Filter

- Use ITS native attributes
- Add `translate="yes|no"` if the annotation is generated when the target language of the document is known.
- In `<unit>` and `<mrk>` elements.

Provenance

- Use ITS native attributes and elements.
- Stand-off elements at the `<unit>` level.
Or should it be at the `<file>` level?
- Applies to the target content: Single instance or reference to stand-off list in `<mrk type="its:any">` element.
- [Example](#)

External Resource

- **Not mapped yet**
- Mapping would likely be related to the Resource Data module.

Target Pointer

- Target of the original document is in the `<target>` elements.

```
selector="//xlf:unit/xlf:source"
```

```
targetPointer="../xlf:target"
```


Id Value

- Use the `name` attribute of the `<unit>` element to store the original ID values
- Using Id Value data category on XLIFF is not really useful as there are no document-wide unique IDs.

Preserve Space

- Use `xml:space` like ITS
- In `<mrk>` and `<unit>` elements
- Note that `<data>` is by default `xml:space="preserve"` while other elements inherit from parents (and `<xlifff>` default is `xml:space="default"`)

Localization Quality Issue

- Use ITS native attributes
- In `<mrk>` elements, with stand-off notation at the `<unit>` level.
- [Example](#)

Localization Quality Rating

- **Not mapped yet**
- In some aspects similar to `<match>`'s `matchQuality` (which is mapped to MT Confidence)
- But has two representations: a score and a number of votes, so using the native ITS attributes may be simpler

MT Confidence

- In the Translation Candidates module:
 - Use `matchQuality` (scaled to 0.0-100.0)
 - In `<match>` element
- Normal inline content:
 - Use ITS native attributes
 - In `<mrk>` element
- [Example](#)

Allowed Characters

- Use ITS native attributes
- In `<mrk>` element
- [Example](#)

Storage Size

- **Not mapped yet**
- To map with the Size and Length Restriction module

A few links

- ITS 2.0 Specification

<http://www.w3.org/TR/its20/>

- XLIFF 2.0 Specification

<http://docs.oasis-open.org/xliff/xliff-core/v2.0/xliff-core-v2.0.html>

- ITS 2.0 mapping for XLIFF 2:

http://www.w3.org/International/its/wiki/XLIFF_2.0_Mapping

- Okapi XLIFF Toolkit (implements the mapping):

<https://code.google.com/p/okapi-xliff-toolkit/wiki/ITS>