

XLIFF 2.0 and Metadata

Yves Savourel – ENLASO

Localization World – October 2012 - Seattle

XLIFF 2.0 in a Few Words

- A simple “core” vocabulary
- Additional standard modules
- Custom extensions
- Processing requirements

- Cross-platform validation tool
- Cross-platform toolkit with library

```
<xliff version="2.0" xmlns="urn:oasis:names:tc:xliff:document:2.0"
  srcLang="en-us">
  <file>
    <unit id="tu1">
      <segment>
        <source>Hello <pc id="1" nidEnd="d2"
          nidStart="d1">World</pc>!</source>
      </segment>
      <originalData>
        <data id="d1">&lt;b></data>
        <data id="d2">&lt;/b></data>
      </originalData>
    </unit>
  </file>
</xliff>
```

Example of simple XLIFF 2.0 document

What is Metadata?

- Basically: Data about the data
- In XLIFF:
 - the **data** is the extracted and translated content.
 - The **metadata** is the information associated with the content (type of an inline code, state of a translation, original ID of a text unit, etc.)

“Integrated” Metadata

- Metadata that is part of the core

For example:

- Type of inline codes
 - Status of a segment
 - Indicator of similarity between a match and the source entry
 - Etc.
- Interoperable

Custom Metadata (Extensions)

- With the XLIFF Metadata module
- With non-XLIFF namespaces
(user-defined or other standards like ITS)
- In both cases:
 - Can be placed only at specific and limited extension points
 - Associated with processing requirements
 - Have limited interoperability

Processing Requirements

- A user extension MUST NOT provide the same functionality as an existing XLIFF core or module feature, but only complement the XLIFF features.
- Merging tools MUST NOT rely on user extensions, other than the ones possibly defined in <skeleton>, to create the translated version of the document.
- User agents that do not support a given user extension MUST preserve, without modifications, that extension (except when the element where the extension is located is removed).

Metadata Module

- `xmlns:m="urn:oasis:names:tc:xliff:metadata:2.0"`
- `<m:metadata>` element with a list of `<m:meta>` elements
- `<m:meta>` has one attribute: `type`
Its value is unique within the `<m:metadata>` element
(→ can be mapped to a hash table)
The content of `<m:meta>` is the metadata

Metadata Module - Example

```
<unit id="1">  
  <segment>  
    <source>Some text</source>  
  </segment>  
  ...  
  <m:metadata>  
    <m:meta type="xyz:forbiddenPatterns"  
      >(\A[^\<]*?>)|(<[\w!?!?/].*?(>|\Z))</m:meta>  
  </m:metadata>  
</unit>
```

Custom Namespaces

- Simple elements, complex elements, attributes
- Private namespaces, other standards or common namespaces

Examples: ITS Metadata

- ITS 1.0 (Internationalization Tag Set) is a W3C Recommendation
- Used to markup the source document with internationalization and localization-related metadata (e.g. localization notes, terminology identification, etc.)
- ITS 2.0 under development, adds many new localization-related information

Examples: ITS Metadata

ITS 2.0 provides metadata for XML and HTML5 that can be:

- Set in the source document and carried over into the translation format
- Added during the localization process

The representation of ITS information in XLIFF is a work in progress:

(http://www.w3.org/International/multilingualweb/lt/wiki/XLIFF_Mapping)

Demonstration

- Extraction done with Rainbow which uses the library of the XLIFF Toolkit
- The **XLIFF Toolkit**:
 - Developed as part of the Okapi Framework
(but hosted separately and without dependency to other Okapi libraries)
<http://code.google.com/p/okapi-xliff-toolkit/>
 - Cross-platform Java library
 - Free and open-source (LGPL)
 - Available with Maven or in ZIP distribution

References

- Read the latest XLIFF 2.0 Editor's Draft:
<https://wiki.oasis-open.org/xliff/>
- Comment or ask questions in the mailing lists:
<https://lists.oasis-open.org/archives/xliff-comment/>
<https://lists.oasis-open.org/archives/xliff-users/>
- Try out the toolkit:
<http://code.google.com/p/okapi-xliff-toolkit/>