TMX Wish List

FEISGILTT 2016

Dublin

Angelika Zerfaß

TMX usage

- TMX is being used widely to move data between systems and so far it does a good job.
- But when asked what TMX should be able to do on top of its current capabilities, the answer often is…
  - Save as much information as possible!
  - Make it more of a standard.

Agenda

- Comments on translation units
- Change history of translations/translation units
  - Rollback to previous version
- Where does the TU come from (import from what TMX file / added manually)
- Segmentation information
- Stricter definition for formatting tags
- Stricter definition of TMX structures
- Exchange of context data

Comments on translation units

- Users would like to be able to save comments that they have made to a segment in the translation editor also to the translation memory.
- The comment could define when a segment should be used/not used
  - Ex: Use "ProductName®" in a heading, use "ProductName" without trademark symbol in the rest of the text.
  - Use this abbreviated segment when it comes up in a table and use the longer non-abbreviated segment when it comes up in regular text or as a heading

Change history of translations/translation units

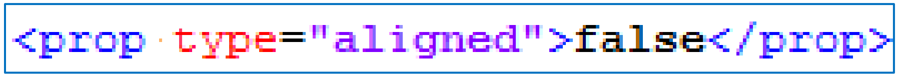
- A translation unit shows the date when it was created and the date when it was last saved/changed.
- Users would often like to know what changes have been made to the unit.
  - If the history of a translation or changed source segment is saved, translation tools could offer a rollback feature to revert to a previous version of the segment.
  - Discussions with clients who change translations back and forth could be shortened (especially when it comes to the question who is going to pay for implementing the changes).

Where does the TU come from

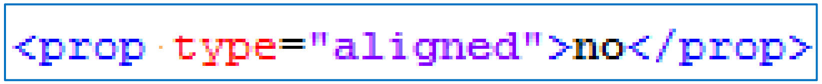
- A translator very often decides whether to use a translation or not based on the meta information associated with the TU.
  - Who confirmed it last
  - Does it come from an alignment
  - The information if it came from an import might be useful as well (and the user might then have the opportunity to check during import of a TMX file whether the same segment from the same source has already been imported)

Where does the TU come from

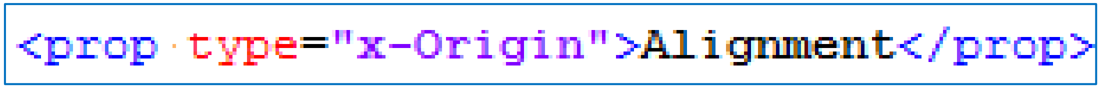
- Most tools have a way to include the information whether a segment came from an alignment (to apply penalties to such segments).
- But this information is mostly not transferable to other tools.



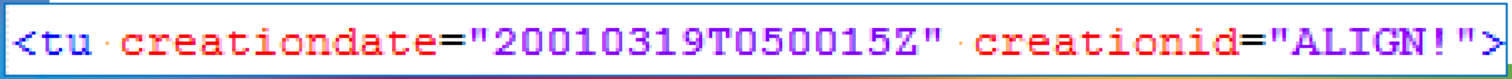
```
<prop type="aligned">false</prop>
```



```
<prop type="aligned">no</prop>
```



```
<prop type="x-Origin">Alignment</prop>
```



```
<tu creationdate="20010319T050015Z" creationid="ALIGN!">
```

Segmentation

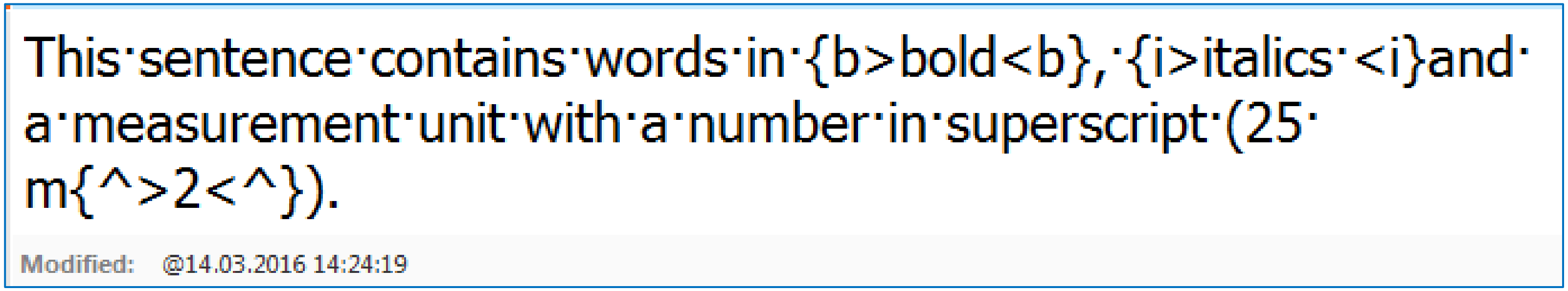
- Although there is a way to include the information what segmentation rules are set in a tool through the SRX format, this does not show how the individual segments were created, but only what segmentation rules were set when the TMX file was exported.
- Segmentation information on the segment level might help TM tools to do a better job with matching (especially if segments have been manually joined or split by a user or when abbreviation lists have been used.)

Stricter definition for formatting tags

- Although TMX allows different ways to code formatting information, some users of TMX use their own codes – which will show up as pure text for users of other tools. In this case, TMX is not a standard exchange format any longer.

Stricter definition for formatting tags

- Coding formatting information not as TMX inline tags but with self-defined codes.



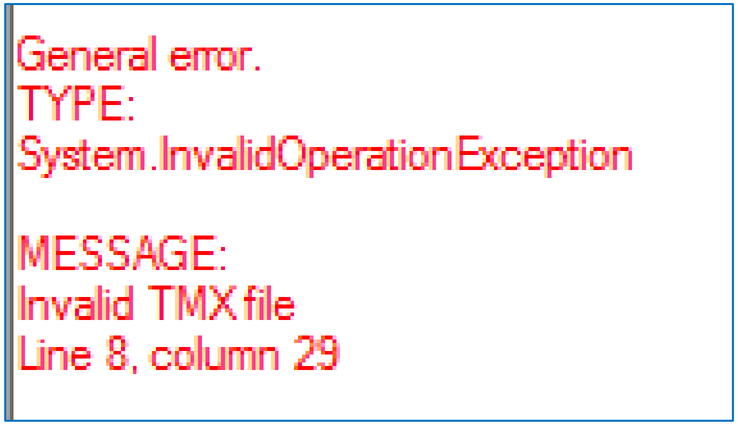
This·sentence·contains·words·in·{b>bold<b},·{i>italics·<i}and·a·measurement·unit·with·a·number·in·superscript·(25·m{^>2<^}).

Modified: @14.03.2016 14:24:19

- Exchange with other tools creates lower matches than expected.

Stricter definition of TMX structures

- A recent example of a TMX file that did not import into a translation tool, because the tool finds the TMX file invalid:

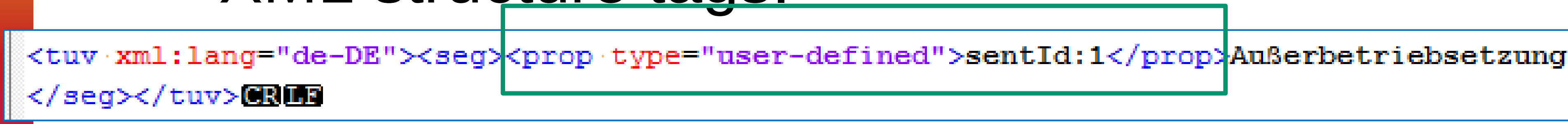


```
General error.
TYPE:
System.InvalidOperationException

MESSAGE:
Invalid TMX file
Line 8, column 29
```

Stricter definition of TMX structures

- The tool does not like the placement of a metadata tag inside the segment tag pair. Between the <seg> tags there should only be the actual segment text, no more XML structure tags.



```
<tuv xml:lang="de-DE"><seg><prop type="user-defined">sentId:1</prop>Außerbetriebsetzung
</seg></tuv>CRLF
```

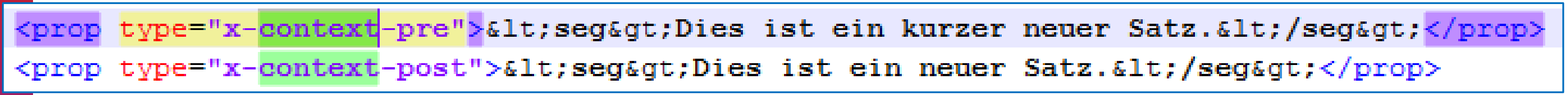
- After deleting the <prop> tag, the TMX file imported without issues.

Exchange of context data

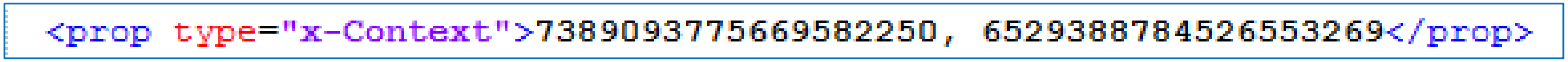
- Most translation memory tools save some kind of context information with the segment pairs.
- This allows for more precise matching and less checking whether 100% matches really appear at the correct position.

Exchange of context data

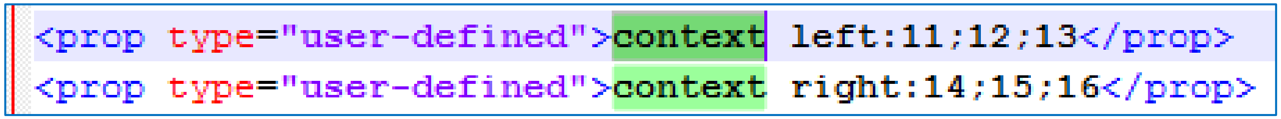
- Different context representation
  - Actual segment text before and after the source segment
  - Hash code (created of previous segment, structure type of the segment etc.)
  - Context numbers



```
<prop type="x-context-pre">&lt;seg&gt;Dies ist ein kurzer neuer Satz.&lt;/seg&gt;</prop>
<prop type="x-context-post">&lt;seg&gt;Dies ist ein neuer Satz.&lt;/seg&gt;</prop>
```



```
<prop type="x-Context">738909377566958225O, 6529388784526553269</prop>
```



```
<prop type="user-defined">context left:11;12;13</prop>
<prop type="user-defined">context right:14;15;16</prop>
```

TMX or XLIFF?

- With the wish list for TMX the question comes up whether it is time to create TMX 2.0 or whether XLIFF might be the better way to go.
- Yes, TMX files can be multilingual and XLIFF are bilingual – but there are not really that many multilingual TMX files around ☺…

One last wish
TMX Editor

- Update Olifant, please ☺ and add some features like
  - Adding lines between TUs so that a user can split up segments
  - Highlighting text you search for
  - Sorting (for example by segment length)
  - Regular expressions in the search dialog

What do you think?