



XLIFF for alignment pairs and segment version history in memoQ

Gergely Vándor
Kilgray Translation Technologies
gergely.vandor@kilgray.com

Who, what, where

Kilgray: translation technology company, develops a TENT called memoQ, a server for it, and several other language tech products

Down-to-earth approach to standards: we try to do what is good for our users

We use XLIFF in some unorthodox ways: alignment and version history (in addition to the basic document exchange functionality)

We would absolutely like to hear what you think about this!

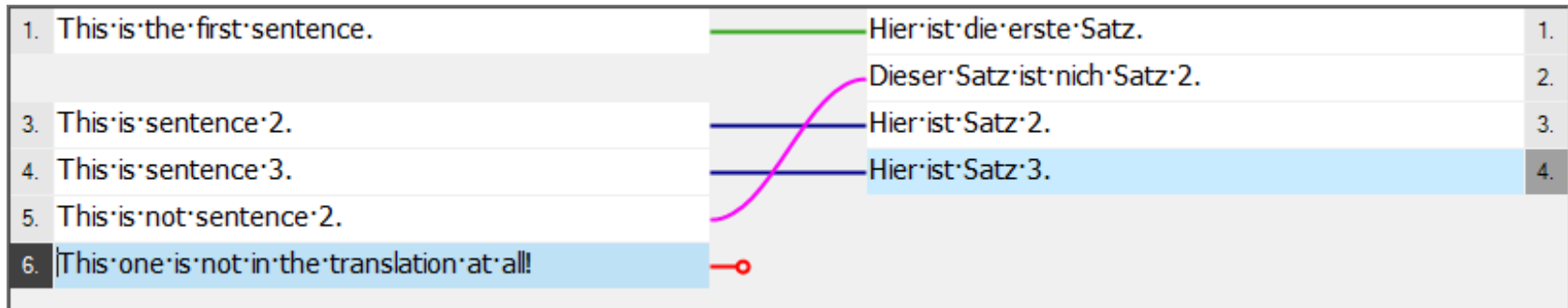




Alignment and version history: what you can do now

- It all exists: all the XLIFF related features discussed in this presentation are ready in the current memoQ version
- XLIFF extensions are used to make them possible
- memoQ users can exchange alignments pairs, and translation documents with version history
- Because of the extensions, direct exchange with other tools not directly possible

Alignment: how it looks in memoQ



1. This is the first sentence.	Hier ist die erste Satz.	1.
	Dieser Satz ist nich Satz 2.	2.
3. This is sentence 2.	Hier ist Satz 2.	3.
4. This is sentence 3.	Hier ist Satz 3.	4.
5. This is not sentence 2.		
6. This one is not in the translation at all!		

The screenshot shows the memoQ alignment interface. It consists of two columns of text. The left column contains source sentences, and the right column contains target sentences. Each sentence is preceded by a number. The sentences are: 1. This is the first sentence., 3. This is sentence 2., 4. This is sentence 3., 5. This is not sentence 2., and 6. This one is not in the translation at all!. The target sentences are: Hier ist die erste Satz., Dieser Satz ist nich Satz 2., Hier ist Satz 2., and Hier ist Satz 3. The alignment is visualized by lines connecting the source and target sentences. A green line connects '1. This is the first sentence.' to 'Hier ist die erste Satz.'. A pink curved line connects '3. This is sentence 2.' to 'Hier ist Satz 2.'. A blue line connects '4. This is sentence 3.' to 'Hier ist Satz 3.'. A red line connects '6. This one is not in the translation at all!' to an empty row in the target column. The thickness of the green line indicates confidence. The blue line indicates manual confirmation. The pink line indicates a cross-link. The red line indicates an insertion.

- Green: automatic by memoQ, thickness shows confidence
- Blue: manual, confirmed by the user
- Pink: cross link, different order in source and translation
- Red: insertion, only exists on one side

Alignment in XLIFF export by memoQ

```
<trans-unit id="1" mq:lastmodificationtime="2011-09-27T22:18:12.8319192" mq:confidence="0.1088867" mq:type="Auto">
  <source mq:segpart="1">This is the first sentence.</source>
  <target mq:segpart="1">Hier ist die erste Satz.</target>
</trans-unit>
<trans-unit id="2" mq:lastmodificationtime="2011-09-27T22:18:56.3953166" mq:crosspair="5" mq:type="User">
  <source/>
  <target mq:segpart="2">Dieser Satz ist nich Satz 2.</target>
</trans-unit>
<trans-unit id="3" mq:lastmodificationtime="2011-09-27T22:18:43.0855424" mq:type="KeepOrderUser">
  <source mq:segpart="none">This is sentence 2.</source>
  <target mq:segpart="3">Hier ist Satz 2.</target>
</trans-unit>
```

- Each row of the grid is one trans-unit here
- Source or target may be empty if empty in that row
- When there is a cross link, the ID of the pair tells which line
- Link type, confidence also in mq: extensions



Version history in XLIFF

- Standard approach: store the current state of the document in XLIFF
- memoQ can also store its history, or the audit trail: exactly who did what when – can be useful or even vital
- You can review this in memoQ, or use external software to access and process it – endless possibilities

Version history – how it works in memoQ

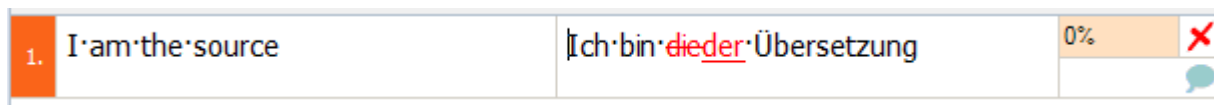
- Versions apply to documents (not segments)
- All changes of a translation row (trans-unit) tracked

1.	I am the source	Ich bin der Übersetzung	0%	✖
				💬


- i.e. source, target, status, metadata, comment...
- Versions stored automatically when something relevant happens in the workflow: like the translator delivers
- But the user can create a snapshot manually as well

Version history – how it works in memoQ

- Differences between any two versions can be seen, track changes style



- Individual rows can be checked for complete history

		Target segment	Version	Date	User
		Ich bin der Übersetzung	Current (modified)	2011.09.27. 1:53	G
		Ich bin die Übersetzung	1.1	2011.09.27. 1:53	G

- Or the whole thing can be exported to XLIFF, for full access to the audit trail!

Version history – in XLIFF exported by memoQ

- Information about a document version, why it was created, when, by whom (as extensions in the XLIFF <header>):

```
..... <mq:minorversioninfo mq:minorversion="1" mq:comment="" mq:createdthroughview="false"  
mq:createreason="Snapshot" mq:creationtime="119273127" mq:creatoruser="G" mq:tag="">  
..... <mq:details/>  
..... </mq:minorversioninfo>
```

- One previous version of a segment (as extensions in a <trans-unit>), incl. source, target, status, etc.

```
..... <mq:historical-unit mq:minorversionend="1" mq:minorversionstart="1" mq:status="PartiallyEdited"  
mq:segmentguid="87995191-a661-48fe-b82c-6abbb735c45d" mq:translatorcommittimestamp="0001-01-01T00:00:00Z"  
mq:reviewer1committimestamp="0001-01-01T00:00:00Z" mq:reviewer2committimestamp="0001-01-01T00:00:00Z"  
mq:lastchangedtimestamp="2011-09-27T23:53:56Z" mq:maxlengthchars="-1" mq:nosplitjoin="false">  
..... <source mq:segpart="1" mq:hasfollowingobject="hasfollowingobject">I am the source</source>  
..... <target>Ich bin die Übersetzung</target>  
..... </mq:historical-unit>
```

What do you think?

- Do you think these uses of XLIFF are:
 - exciting ideas, and may affect XLIFF itself?
 - unacceptable hacks against the spirit of XLIFF?
 - boring, as you have thought of all this already?



Approach to standards for SW vendors

- Be very good citizens and build the SW around the current state of standards like XLIFF?
- Or always revolt and upturn things: the HTML case
- Ideal is probably somewhere in between: HTML is successful but very messy technically...
- Kilgray wants to be both progressive/experimenting and respectful



But... why XLIFF for these at all?

- Openness: we will publish the schema, and techies can read these special XLIFF files
- Alignments are closely related to bilingual documents for the end user in memoQ, and we didn't want to add formats
- Translation documents: we want our XLIFF export to be a full representation of documents, this includes versions
- Possibility of integration with other systems, and access the memoQ audit trail



Hurdles: what we wanted to do but couldn't

- When reimporting a document, a new version store is created for it (after 1.7, you add 2.0 and start over)
- Several version histories for each document, this proved too much to be shoved into one XLIFF file
- So only the last active version store is saved to XLIFF, you get 2.0, and 2.1, etc, but you lose everything in 1.x
- This is sad, because it is against our principle of full document representation

... and Thank you!

Questions?