FEISGILTT 2016 XLIFF OMOS Public Info sessions

Draft meeting minutes         Tue June 7 and Wed June 8, 2016

Participants:
[participation in particular sessions was not tracked, this list is based on FEISGILTT 2016 participation]

XLIFF OMOS TC members
Chair
David Filip (ADAPT Centre)

Secretary
Loïc Dufresne de Virel (Intel),

Members
Ryan King (Microsoft), Kevin O’Donnell (Microsoft), Soroush Saadatfar (ADAPT Centre @ LRC @ University of Limerick), Felix Sasaki (Individual), Phil Ritchie (Vistatec), Chase Tingley (Spartan Software)

Public
Victor Alves (Individual), Jan Bareš (Moravia), George Bina (Syncro Soft), Konrad Chmielewski (XTRF), David Clarke (Welocalize), Mariza Flores (Thousand Words Translations), Andreas Galambos (Transmission Übersetzungne), Andrew Gibbons (Welocalize), Ján Husarčík (Moravia), Łukasz Kaleta (XTRF), Gary Lefman (Cisco), David Lewis (ADAPT Centre), Ian Morris (Individual), Dimitris Orfanoudakis (TAUS), Achraf Oumghar (Lionbridge), Felix Sasaki (DFKI), Uta Seewald-Heeg (Anhalt University of Applied Sciences), Richard Sikes (Content Rules), Vinod Sudharshan (TAUS), Angelika Zerfaß (zaac).

XLIFF OMOS TC Public Info Sessions started at 1.30pm WEST on June 7th 2016.

XLIFF OMOS Debrief

Chair: dF

Panelists: XLIFF OMOS TC membership
dF debriefed the delegates on the XLIFF OMOS TC formation at OASIS https://www.oasis-open.org/news/announcements/call-for-participation-xliff-object-model-and-other-serializations-xliff-

Charter of the XLIFF OMOS TC [https://www.oasis-open.org/committees/xliff-omos](https://www.oasis-open.org/committees/xliff-omos) has been reviewed

In particular [https://www.oasis-open.org/committees/xliff-omos/charter.php](https://www.oasis-open.org/committees/xliff-omos/charter.php)


dF explained how it was necessary to form the new TC because the new standardization items requested by the XLIFF community in Dublin and Vancouver 2014 were out of scope of the original XLIFF TC Charter (https://www.oasis-open.org/committees/xliff/chartePHP.php) albeit clarified and updated.

stressed the importance of the Non-Assertion IPR mode for open source reference implementations of JLIFF and any APIs to be based on the XLIFF OM.

**XLIFF OM Work in Progress**

*Chair: dF*

*Panelists: XLIFF OMOS TC membership*

dF explained about the OMOS TC plan to work on git as a version control. Since OASIS hasn’t yet launched git as a version control option for TCs, dF agreed with OASIS Admins that a temporary GitHub repository [https://github.com/DavidFatDavidF/XLIFFOM](https://github.com/DavidFatDavidF/XLIFFOM) will be used with a Readme vetted by the OASIS Admins.

dF explained that the UML class diagram that the TC has been discussing [https://www.oasis-open.org/apps/org/workgroup/xliff-omos/email/archives/201605/msg00002/xliff_OM_v4.png](https://www.oasis-open.org/apps/org/workgroup/xliff-omos/email/archives/201605/msg00002/xliff_OM_v4.png) is being redeveloped with amendments as the discussion progresses on the GitHub repository.

The panel agreed that the group and unit data items were distinct entities not only in the XML serialization but also in the abstract data model.

Ryan was arguing that segment and ignorable were actually only facets of a single subunit data item. The panel actually reached consensus that it was indeed the case and not a JSON driven artifact. The resulting action item for dF and Chase and whoever is going to participate on the XLIFFOM repository is to introduce segment and ignorable as facets of a single subunit data item.
dF explained that the subunit or inline data model within the XLIFFOM is fully linear not allowing for XML-like recursion. OMOS TC previously agreed that `<pc>` and `<mrk>` were XML specific and not needed in the abstract model.

Andrew Gibbons (Welocalize) suggested that information on XLIFF using inline pair tags will get lost in a transform facilitated by the XLIFFOM. dF agreed that this was the case, however the pair tags have an exhaustively defined equivalence in XLIFF 2 and XLIFF Modifiers were free to transform pair based inline notations into atomic marker based notations and vice versa (if possible WRT XML well-formedness constraints). Most importantly, **Mergers** were obliged to accept pair or atomic inline tags disregarding
whether the *Extractor* had actually used pair or atomic representations during the initial *Extraction*. dF also said pooling/joining of all segments and ignorables within each unit was a recommended best practice for *Mergers*. *Mergers* are obviously allowed to perform equivalent modifications when accepting *XLIFF Documents* after a roundtrip.

**JLIFF 1.0 Work In Progress**

Chair: dF

**Reporter: Ryan King (Microsoft)**

Ryan reported on the JLIFF examples generated between Yves and himself.

Ryan explained how JLIFF was just another serialization and the true challenge is to make sure that both XLIFF and JLIFF share the same data model (OM).

[Ryan’s presentation will be uploaded]

*XLIFF OMOS TC Public Info Sessions were adjourned on 7th June 2016 at 4pm WEST*

*XLIFF OMOS TC Public Info Sessions started at 1.30pm WEST on June 8th 2016.*

**1st TMX Symposium (Is this the time for TMX Next?)**

Chair: dF

**Reporter: Angelika Zerfaß (zaac)**

The TMX Symposium started with Angelika’s industry wish list presentation:


This gave to the delegates the idea what is the TMX use case and what were the regular user’s frustrations with the current version of the standard and a variety of its implementations.
Dave Lewis introduced what features would be needed to make TMX based data useful as Linked Open Data


Most importantly, TMX has no explicitly defined fragment identification mechanism, which see the major drawback in making TMX stores work as LOD.

Discussion (Workshop: Is this the time for TMX Next?)
dF explained that the XLIFF OMOS TC was chartered, so that TMX would fall within its scope. The chief reason for the Chartered scope to allow for further development of TMX was the fact that ETSI ISG LIS that came to own TMX after the LISA demise in 2011 was disbanded in late 2015. However, OASIS XLIFF OMOS will not be legally entitled to perform any work on maintenance or continued development of the TMX standard until OASIS has negotiated the IP transfer with ETSI. The status of this negotiation is unclear. It’s even been impossible to find out if the negotiation started to date.

Based on discussing the position statements of the reporters, the following strengths, standardization gaps and issues and with the current TMX version and implementations were identified.

**Strengths**
- Multilingual – good for post mortem bulk exchange
- Widely adopted
  - However falling short of the lossless leverage promise due to incompatible implementations (see implementation issues)
- Storage efficient (unordered, doesn’t store context)
  - This strength was perceived as outdated with the steep drop of storage pricing that happened since 2004.

**Missing features:**
- Common explicitly defined fragment identification mechanism
  - Parts of TMX files cannot be reliably referenced
- Context matching, either proper context or id based – possible issues with existing patented IP owned by SDL and Lionbridge, if this gap was to be filled in the next gen standard.
- Insufficient provenance specification
- Mark up salad – hard to support inline data model
  - Need to catch up with the pruned down XLIFF 2 inline data model
- Even if implementers were sharing SRX rules along with their TMX files, the TMX standard lacks any re-segmentation provision, which is probably hardly possible with the basic design principle of an unordered segment pairs collection.
Implementation issues

- Most of the implementations are not Level 2 compliant, as result implementations are not able to properly exchange segments with inline markup, which in turn lowers leverage that can be achieved by TMs exchanged via TMX. See also Mark up salad.
- Implementers don’t implement SRX along with TMX, as originally intended by LISA OSCAR, as result the receiving implementer don’t know what kind of segmentation rules was applied when creating the TMX, which in turn lowers leverage.
  - Some implementers blatantly disregard the TMX inline markup provisions and use their own proprietary non-XML markup that cannot be even recognized as markup by other implementations, thus farther damaging the exchanged leverage (based on implementation cases reported by Angelika).
    - However, this is something not addressable by standardization proper. It’s a mindset and evangelization issue. Implementers simply cannot setup their own secret markup within payload of an exchange format spoiling the game for everyone else.
- Lack of a powerful state of the art TMX editor
  - Olifant still the best usable tool available
  - Need better filtering and analytic capabilities

Possible solutions

1. Develop TMX Next at XLIFF OMOS TC
   - Blocking issues:
     - OASIS need to conclude TMX IP transfer negotiation with ETSI. Outlook unclear.
   - Possible issues down the road:
     - Need to make sure that Context Matching IP owners are represented. Lionbridge currently not on the TC. The IP owners would either need to provide their Non-Assertion Covenants to protect the TMX Next implementers, or the IP would need to be worked around.
   - Advantages:
     - Clear upgrade path, TMX next would be a clear successor of TMX 1.4b (1.4.2)

2. Develop an XLIFF 2 profile that would fully replace TMX in TM exchange scenarios
   In the XLIFF Symposia (held since 2010, as part of FEISGILTT since 2012) it transpired that some tool makers and service providers were using XLIFF as a leverage facilitating exchange format to overcome TMX related issues and lack of TMX development since 2004 or even 2001.
   Several delegates suggested that XLIFF can be used for TM exchange purposes, since the storage efficiency advantage of TMX has become obsolete since 2004 due to the continued price drop of storage capacity. However, the need to duplicate the source language with each new XLIFF file per target language seemed prohibitive from the point of view of transfer bandwidth.
Ryan King (Microsoft) suggested that the reference language mechanism available through the translation candidates module should be used to fully replace TMX as multilingual TMX exchange format.

After a short discussion, mainly between Ryan and dF, it was agreed that such a profile can be made fully XLIFF 2 conformant. The main aspects being:

1. The TM exchange XLIFF MUST contain all source sentences in core <source> elements
2. The trgLang attribute MUST NOT be defined (hence no core <target> elements in the file).
3. All <mtc:match> elements REQUIRE the OPTIONAL reference attribute to be set to “yes”.
4. Each of the <target> children of all the <mtc:match> elements would REQUIRE their OPTIONAL xml:lang attribute to be set to the BCP 47 tag value corresponding to one of the target languages of the multilingual TM.

- **Blocking issues:**
  - N/A

- **Possible issues down the road:**
  - The status of TMX as a legacy format could not be clearly addressed.
  - Possible change management issues when pushing the new exchange format adoption.
    - Nevertheless, Angelika said that her customer stakeholders don’t care what the TM exchange standard is called (they don’t necessarily know that it is currently TMX) provided that they do not lose leverage due to the exchange vehicle.

- **Advantages:**
  - Shared inline data model with XLIFF 2 for free – hence no leverage loss due to incompatible inline markup
  - Re-segmentation possible
  - TMX transfer negotiations between OASIS and ETSI could be dropped.
  - Would not infringe on SDL and Lionbridge context matching IP.

**XLIFF OMOS TC Public Info Sessions were adjourned on 8\textsuperscript{th} June 2016 at 3.50pm WEST**