Language Interoperability Portfolio (Linport) Project

A Standard Container Format for the Localization Industry

PROJECT OVERVIEW & HISTORY

Why a Container? Issues

- Breakdown of translation costs:
  - ~50%: management and overhead
  - 30%: translation
  - 20%: profit
- Manual tasks prevent automation, limit growth
- We have maxed out our current model: "Beyond human scale"
- Lack of metadata leads to substantial costs (large LSP reports ~$3.5 million/annum)
- Tools lock-in

Why a Container? Issues

- Quality undefined, but suffers…
- Speed and scalability: We have reached the point where we cannot add more bodies
- Increased dependency on tools
- Integration of multiple translation modalities (MT, HT, CAT, TM…)
- If we do not change, we *cannot* meet future demands

Linport: Where We Are…

Linport: Where We Need to Be…

Image: melted_snowball

Structured Specifications

- Why not just data interoperability?
- What do the specifications cover?

ASSUMPTIONS, PRINCIPLES & ISSUES

Assumptions - 1

- A package can be transmitted either as a single self-contained zip file or a zip file consisting primarily of URLs to external items.

Assumptions - 2

- This format should keep the entire ATP (authoring, translation/localization, publication) multilingual document production chain in mind.

Assumptions - 3

- The types of payloads for this format will include source texts and resources for the translator, such as glossaries. Bilingual files (source text and translation, aligned) conforming to a profile of XLIFF are also a desired payload for the format. The scope of the Linport project does not include the development of payload formats, but it may involve specifying preapproved standard payload formats in a strict version of Linport.
- It does include cooperation with groups that are developing payload formats.

Assumptions - 4

- This format should be based in part on existing tool-specific package formats.

Assumptions - 5

- This format should take into account any existing formats used within government agencies concerned with translation, such as the EU Translation Centre, the US NVTC, and the Canadian BTB.

Assumptions - 6

- This format should include slots for Structured Translation Specifications (STS) from the forthcoming ISO TS 11669 (see www.ttt.org/specs for a list of the parameters that are used to structure specifications).

Assumptions - 7

- The format should be standardized in three phases: (a) a "blueprint" for early implementers; (b) an industry standard within Oasis and/or ETSI, and (c) an ISO TC 37 standard.

Assumptions - 8

- The documents describing this format should be available as a free download with a Creative Commons attribution license
- Any reference implementation software should freely available, including source code, royalty free, and with multiple open source licenses to choose from.

Assumptions - 9

- There should be reference software that allows anyone to build a package (even without any particular translation tool), view a package, and edit a package.

Assumptions - 10

- Competition among package formats from multiple projects should be avoided.

Principles - 1

1. Develop an abstract data model that can be represented using a choice of structural styles.
2. Define profiles of the main data model (e.g. a profile for a simple bilingual translation project)
3. Develop reference software in parallel with documents describing the format, not after the fact.

Principles - 2

4. Encourage a few tool vendors to be early implementers and provide feedback.
5. Encourage a few content owners to be early implementers.
6. Use agile development methods.
7. Strive for simplicity (so the format is easy for tool vendors and content owners to implement).

Principles - 3

8. Allow for package validation on mulitple platforms.
9. In initial versions, do not include workflow information except for package id and person to contact if there are questions.
10. In initial versions, focus on translation/ localization, but connect with content management systems as soon as feasible and interact with authoring.

Issues - 1

- **Human legibility**. Should a package include something like an index.html file that facilitates human legibility using only a browser without an Internet connection? Or should the there be a separate application for viewing a package (either Web-based or standalone)? Or should should inspection of a package by humans be discouraged?

Issues - 2

- **Human editing**. Should humans be encouraged to edit a package without any tools other than an unzip utility and a text editor? This a very different issue than read-only human legibility.

Issues - 3

- **Structural styles**. Should the abstract data model be represented using XML, key-value pairs (Xdossier style), or some other structural style?

LINPORT DATA MODEL OVERVIEW

BTS DATA MODEL OVERVIEW

BTS BUILDER DEMO

FUTURE ACTION

FINAL REMARKS & ADJOURN

Funding

- Contributions needed to support development
- Ideas for additional services/funding methods?