The Container Project: Towards an Open Nonproprietary Format for Packaging Translation Materials

Alan Melby, ArleLommel, Nathan Rasmussen and Jason Housley

4051 JFSB Brigham Young University, Provo, UT 84602, USA

akmtrg@byu.edu, arle.lommel@gmail.com, volodymyr.velyky@gmail.com, housleyjk@gmail.com

Abstract. ISO standards for intermodal shipping containers have dramatically improved efficiency within the shipping industry worldwide. The translation/localization industry needs an analogous standard for translation tasks. There are a variety of proprietary translation packets that allow materials relevant to the translation project (the source text, various resources such as translation memory files, etc.) to be put into one or morepackets and sent to a translator. The translator can then use the same format to return the requested translation. The objective of the Container Project is to define an opennonproprietary format for creating translation packets. An important feature of the Container Project is structured translation specifications compatible with the system of specifications in ISO TC 37 WD 11669 project (see ISO TC37/SC2/WG6).

Keywords: translation, localization, interoperability, standards, authoring, tools

1 Introduction

One of the consistent problems faced by the translation and localization industry is that material to be translated can be transmitted in many different fashions, often with incomplete or inadequate instructions for how the project is to be completed. In many ways this is analogous to a problem faced by the shipping industry 20 years ago, when goods were shipped in a number of different containers [1]. Using multiple kinds of containers to ship various goods meant that teams of workers were needed to load cargo from one container to anotherwhen the method of shipping changed, e.g., when material was moved from a ship to a train, making transportation of goods very expensive in real terms.

The development of ISO standards for the construction and marking ofintermodal shipping containers (ISO 688, 790, 1161, and 1897) has helped to alleviate this problem because a standard container can now be used to transport goods by plane, truck, train, or boat. The actual content inside the container might be very different from the content of other containers, but because the containersare standardized, materialsin them can be transported by any vehicle designed to use the relevant ISO standards without manual unloading and reloading of containers. As a result, real costs for freight shipment have declined greatly and shipment volumes have increased dramatically.

Similar to the days before ISO standards for shipping containers, translation service providers (TSPs) currently spend a significant amount of manual effort in manipulating files, clarifying instructions, and verifying that files move from place to place correctly and are translated according to the client's expectation. For example, a freelance translator may be contracted via email to translate marketing survey responses that are stored in an online repository. Instructions for accessing the repository are sent separately from any reference materials, which are given in later emails. To interact with quality control personnel, the translator is asked to correspond through an instant messaging service. Thisapproach means that important information pertaining to a project is split between three locations: email, instant messages, and the online repository. This type of system often leads to confusion concerning the details, management, and evaluation of a project. One large TSP recently reported that addressing issues such as clarifying instructions and other cases of manual interventioncost it in excess of 3 million USD per year.

A partial solution to this problem is the use of a standard format for transmitting content,such as XLIFF, which allows localizable material to be extracted and transmitted in a regular fashion. When XLIFF is used consistently, it simplifies the process of obtaining localizable material and ensuring the material's completeness, but it does not ensure that instructions for the translation process are correctly transmitted, nor does it address the needs of all translation tasks (e.g., graphics localization that goes beyond the translation of strings and desktop publishing tasks that are frequently carried out as part of the translation process). So while XLIFF is a tremendous benefit for users, it does not eliminate many of the manual issues associated with the translation process.

1.1 An Open Nonproprietary Translation PacketFormat

The need to package all of the materials necessary for a translation project (the source text and any terminology files, translation memory files, reference materials, etc.) in one place is so clear that many of the leading translation environment tools already provide a format for bundling translation materials. However, these formats are proprietary and are generally not compatible with oneanother. Although tool-specific formats allow TSPs to package materials for convenience, other individuals in the chain, from authoring to publishing, may not have access to the same format, leading to a need to again manually handle files. In order to promote interoperability,the translation industry needs a nonproprietary standard format for packaging translation materials that includes full project metadata describing the expectations of the client and TSP.

We propose the development of a standard container or *translation packet*format for translation projects. The use of the translation packet format applies regardless of the tools to be used. The packet provides a standard mechanism for the transmission of translatable content together with other resources needed to facilitate the translation and localization process. The intention is that a complete and valid packet should contain (or reference) all of the materials and project data needed to fully process a transaction, thus minimizing the need for manual intervention or negotiation between the TSP and client after the start of the transaction. The translation packet is intended as a scalable format that is suitable for translation requests of just a few words or of hundreds of thousands of words, using the same generic control structure for all cases.

While the translation packet will not eliminate all manual processing and management steps, it will greatly reduce them by providing a standard way for translation tools to interact with the resources in the packet. Because the packet's structure is flexible but controlled, tools that use it will know how to interpret the contents, meaning that manual intervention would be required only when strictly necessary. (For example, the packet might contain instructions on how to obtain materials at a secure facility, an inherently manual task, but these instructions would persist in the container, eliminating the need to pass separate emails or messages.) The translation packet encourages, but does not require, the use of other standard exchange formats such as TMX, TBX, XLIFF, SRX, etc.

At the heart of the packet is a *structured translation specification set* (STSS), a set of metadata written in XML that explains how the transaction is to take place. The STSS details the client's expectations and requirements for the end translation product. By using the STSS, many of the causes of conflict and redundant or unnecessary work will be eliminated from the translation process and the reasons for any breakdown will be easier to identify. The STSScan also assist in the translation procurement process by identifying in advance variables that are likely to affect project costs and by requiring clients to be clear about what they expect from service providers. The STSS is not an arbitrary set of specifications for describing a translation project; rather, it is a list of parameters that correspond to existing translation quality standards (ASTM F2575-06 as it is updated in ISO TC 37 WD 11669) [2,3]. The goal of the STSS is to accurately describe the translation project at hand, and we postulate that the same parameters can be used to provide an adequate description of nearly every translation project. We also argue that it is possible to develop a standard format for translation packetsthat applies to nearly every translation project, analogous to how intermodal shipping containers can be used to transportmany types of materials [1].

2 Translation Packet Structure

The translation packet itself consists of a ZIP-compressed directorycontaining a pre-defined directory structure (see Fig. 1, overleaf). The ZIP-format specification is open, free, and implemented in all major operating systems [4]. The packet may be optionally encrypted using OpenPGP if security is an issue [5], but the encryption key will necessarily need to be transmitted independent of the packet. The contents of the ZIP-compressed directory are as follows:

1. **Structured Translation Specification File (STSS)** (XML file).
   a. **Global Properties**. The STSS contains a header that includes the unique ID of the packet. It is likely that software will translate the unique ID to a locally chosen, meaningful display name. All other XML files in the

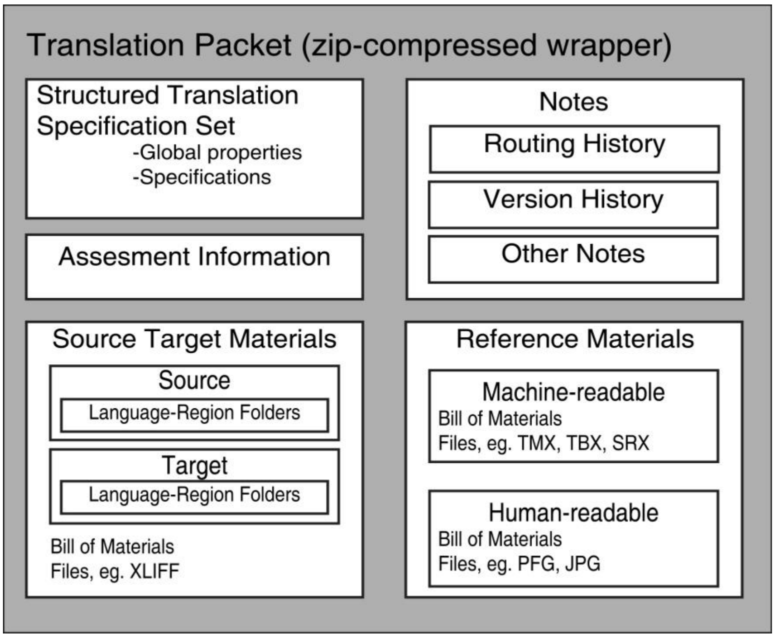

Fig.1. High-level structure of a translation packet

packet must reference the unique ID of the packet to ensure the integrity of the packet. The global properties also include an optional user-defined project ID, which can be used to connect multiple packets belonging to the same transaction or purchase order, etc. The global properties header may also contain contact information for the packet's next destinations.

   b. **Translation Specifications**. For a packet to be considered valid, the specifications must be filled out entirely; incomplete specifications could result in the rejection of a packet or the introduction of additional manual steps that may delay the translation process. See section 3 for a more detailed description of the specifications and their format.

2. **Assessment Information (**XML file**)**. Contains information on quality assessment tasks and requirements.
3. **Notes** (folder). This folder contains three files that provide various auxiliary functions for the packet.
   a. **Routing History**(file). This file provides an audit trail for the packet. Each tool that accesses the file must log its access in the routing history, even if it does not otherwise change the packet. This history is intended for use in automatic processing as verification that the packet has followed proper processes.
   b. **Version History**(file). This file contains a summary of changes made with each processing step. We have sketched a snapshot/rollback function, which would need a file storedhere. We are uncertain whether archival functionality belongs in thepacket itself, and we invite feedback from potential users.

   c. **Other Notes**(XML file). This file contains a log of all transactions involving packet contents. It may also include notes added by humans during processing or use of the packet. It is currently an XML file, but that may be changed to a more human-oriented format in future versions of the packet specification.

4. **Source Target Materials** (folder). This folder contains (or references) the actual materials that are to be translated. The folder contains two subfolders labeled Source and Target. It may also include any XLIFF files or other multilingual sources.
   a. **Bill of Materials** (XML file). All files in the Source Target Materials folder must bereferenced in the Bill of Materials by either a local relative URL (i.e., a path to a file in the packet) or an external URI. The externalURI may be an Internet URL that specifies where the files may befound, or it may be some other identifier. It may refer to files sent in another packet, using a special URN space for this purpose. In thecase of materials that cannot be included or directly accessed (e.g.,files stored at a secure facility or on a server requiring loginvalidation), the external URI may not be resolvable. In such cases theBill of Materials must also contain natural-language instructions onhow to obtain the file. (Note that if files cannot be retrievedautomatically, some tools' automatic processing functions may not be available.)
   b. **Files**. If there are any multilingual files, such as XLIFF, they are included in this directory along with the Bill of Materials.
   c. **Source** (folder). If files that are to be translated are stored locally in the packet, they are included here in language-region tagged folders. An example of language-region tag would be "en-US" for United States English, using ISO 639-1[6] for the language code and ISO 3166[7] for the region. Each file *must* have a corresponding reference in the Bill of Materials.(If the files to be translated reside external to the packet, they will not be included here.) The files to be translated may reside in a deeper directory structure within the language-region specific folders.
   d. **Target** (folder). This folder contains (or references) the actual materials that have been translated. It corresponds to the previous Source folder.
5. **Reference materials** (folder). This folder contains or provides directions to access reference materials to be used in the translation process. It contains two subdirectories:
   a. **Machine-readableReference Materials** (folder). This folder contains or references linguistic reference materials intended to be used in machine processing of the file. These materials may include translation memory (TMX or application-specific files), terminology resources (TBX or application-specific terminology files), segmentation rules (SRX), etc.
      i. **Bill of Materials** (XML file). This file is similar to the other files with the same name(see 4.a and 5.b.i). It may also contain instructions to use existing resources (e.g., to use translation memory data that the TSP already has in-house). If these existing resources are contained in another packet, the resources can be

retrieved and loaded automatically given a project ID or the unique ID of another packet.

ii. **Files**. If any linguistic reference files, such as a translation memory or glossary, are included in the packet, they are found here.

b. **Human-readableReference Materials** (folder). This folder contains or references any materials provided for human translators but that are not intended to be processed in a translation environment tool (i.e., materials that are informative only). Examples include PDFs of source files, samples of similar documents, and background material that may be helpful to the translator.

i. **Bill of Materials** (XML file). See 4.a and 5.a.i.

ii. **Files**. Any local files for human use are to be stored here.

3 Structured Translation Specification Set (STSS)

The core of the translation packet is the STSS. Table 1presents a list of the parameters to be addressed in the STSS. Although in theory the STSS can be constructed by hand, it is expected that the program used to create the packet will request this information and that some of the specifications will be automatically determined. For example, a toolmay perform a word count to obtain the volume of text to be translated.

When dealing with specification sets, a distinction is made between parameter and specification. A parameter is a variable for the translation project; it can be paraphrased as a question. A specification is the descriptive value of a particular parameter; it can be considered the answer to the question posed by the corresponding parameter. For example, a specification for the parameter *file format* might be "a Microsoft Office Word 2007 Document (.docx)."The specifications for one translation project might be very different from the specifications of another project (e.g., translating a patent versus translating subtitles for a movie), but the parameters remain constant. Parameters form a framework for creating structured specifications. Without this framework, the names, descriptions, and order of the specifications for a translation project might vary widely.

Thestructured set of 21 translation parameters listed in Table 1can be broken down into four major groups: *Linguistic*, *Production tasks*, *Translation environment*, and *Other considerations, delivery, and compensation*. *Linguistic* specifications detail information about the source content (its document type, language, intended audience and purpose) as well as target language–specific information. *Production tasks* describe the tasks to be performed as part of the production phase. *Translation environment*specificationsdetail the tools and references that a translator will use. Whereas the first four major groups address the translation project itself, the specifications in *Other considerations, delivery, and compensation* focus on the interaction between the requester and the TSP, as well as allow room for additional clarifications. Many of these parameters are included in national and regional translation quality standards, although they may not all occur together with the same names as those listed here [2, 3].

Table 1. Parameters of a Structured Translation Specification Set (STSS)

A. Linguistic {1–13}

source content–specific information (not dependent on target language)

{1} content type, audience, and purpose
{2} subject field(s) and terms
{3} volume
{4} complexity and obstacles
{5} origin

target language–specific information

{6} language(s)
  a) region(s)
  b) degree of accuracy and fluency
  c) terminology constraints

{7} audience(s)
{8} purpose(s)
{9} content correspondence
{10} usage register
{11} file format
{12} style
{13} layout details

B. Production tasks {14–15}

{14} typical production tasks
  a) pre-translation tasks
  b) initial translation
  c) in-process quality assurance and control

B. Production (cont.)

  i. self-checking/post-editing
  ii. revision
  iii. review
  iv. final formatting or compilation
  v. final reading

{15} additional production tasks
  a) functional testing
  b) back translation
  c) random sampling

C. Translation environment {16–18}

{16} technology
{17} reference materials
{18} workplace-related requirements

D. Other considerations, delivery, and compensation {19–21}

{19} legal and ethical considerations
{20} requirements for TSP selection and terms of delivery
{21} compensation, clarification, and other expectations of the parties

These specifications form a framework that defines and guides a translation project and allows the entire translation project to be evaluated. The first five parameters {1–5} are useful in developing initial project specifications and are highly relevant to pre-production activities. An adequate translator cannot be selected without knowing the content type and subject field(s) of the source material. To estimate the cost of a project obviously requires knowledge of the volume and complexity of the source document. For example, the effort required to translate text in a graphic (e.g., images, diagrams, or even Flash presentations) depends on whether the graphic is available without text or with editable text.Likewise, the number of potential fuzzy or exact matches within a translation memory changes the practical volume of a text. Such source text obstacles may dramatically affect the degree of difficulty in a translation.

The rest of the parameters{6–21} are relevant to quality (i.e., a quality translation is one that conforms to all of the agreed upon specifications). Conformance to some specifications cannot be determined solely by examining the target text. For example, an otherwise good translation that is inappropriately divulged to a third party or that is delivered late would not be considered a job well done.Project specifications are relevant during all phases of a translation project.

Within the STSS, each parameter is labeled with one of three statuses, depending on where the packet is in terms of pre-production negotiations: *incomplete*, *proposed*, or*settled*. An *incomplete* parameter indicates that a specification has yet to be determined, or that the client has no strong opinion about that particular parameter for this project. *Proposed* specifications indicate soft requirements, or that the client is willing to negotiate these details, whereas *settled* specifications are ready to go into production and indicate the hard and fast details of the project. By the time the client and the TSP sign a contract, every parameter needs to have a status of *settled* for the packet to be valid even if the specification is simply “at translator’s discretion.”

The STSS file itself is written in XML. Although wehad initially planned to represent the STSS in a schema compatible with Universal BusinessLanguage(UBL), wefound UBL ill-suited to the needs of the project. Its orientation tominutely subdivided formal data based on existing business practices is apoor match to the STSS because translation specifications requireflexibility in addition to structure. UBL’s complex and abundant technical details arefinely dispersed through its documentation, and tool support to managethis complexity is still too rudimentary for practical use. In light of these considerations we abandoned UBL in favor of a simpler,customized schema.

4 Comparison with Other Formats

Under the rubric *Interoperability Now*[8], several vendors have been developing a translation packet format of their own. While their approachhas much in common with ours, it differs in emphasis and certain details.Interoperability Now mandates the use of a number of open standards such asXLIFF, TMX, and TBX; while we recommend these, we do not prohibitproprietary formats. We have placed great importance on the STSS as a means todocument a shared understanding of how an entire project is to be done; the Interoperability Now format appears to have no analogous formalism and views a packet’s use interms of dispatching a particular step or steps of workflow to anotherparty.

Apart from the STSS, both the Container Project and Interoperability Now anticipate the same types of content in a packet: monolingual content in source and target languages, files containing multilingual data, translation memory, a glossary, and potentially a style guide and other reference materials. Both projects anticipateneeding a standard location to place these materials and will allow URIsto external resources in addition to local storage. Both are considering how to reconcile encryption and confidentiality requirements with interoperability. Both intend to leverage existing open standards. We believe much could be gained from technical exchanges and interaction between the two projects.

One of the Interoperability Now participants, Kilgray, already uses anopen format for handoff packets in its product memoQ [9]. Like our packet,memoQ's packet is a ZIP file, with a prescribed internal format. Designateddirectories hold content in XLIFF, translation memory in TMX, andglossary lists in CSV (a *de-facto* standard though not an official one). Asin our packet, a project identifier and other basic information arecommunicated in an XML file (analogous in part to our STSS and Bill of Materials). Other directories hold data specific tomemoQ, but practical interchange is possible using only these three,and detailed instructions are provided in user documentation. We hopeto advance the interoperability that the memoQ packet provides.

A final, particularly relevant format for comparison is the Multilingual Electronic Dossier (MED). This is a proposal to support interoperability throughout multilingual authoring, translation, and publication. It is now being developed by Manuel Tomás Carrasco Benitez at the Directorate-General for Translation of the European Commission (EC) [10]. We have recently begun a technical collaboration with him. (Note that this collaboration is not policy collaboration, nor do his technical opinions necessarily represent the opinion of the European Commission.)

The MED can be represented as a URI or as a ZIP-file container. Its contents must be discoverable, either through conformity to some predetermined directory structure and filenames, or by being listed in an index file. These contents may be local files, or redirects to other URIs. The MED's internal structure makes use of language codes to help identify the various contents. These similarities to our own proposal arise from a shared goal: a format that allows specialized tools to smooth our workflow but that is still tractable to generic tools.

The MED differs from our project on a few points. It prescribes a syntax for content filenames: name.language.format[.compression]. Its administrative information is Dublin Core metadata, which is very different from the STSS. It has stronger support for concurrent use, when the contents are represented as URIs; unlike the translation packet, the MED does not exclusively own its contents. URI contents are explicitly preferred for use within an intranet, preventing the proliferation of files by writing the data back out in an interchange format only when necessary. An MED should constitute a browsable website for no more effort than unzipping it, dropping it into a Web server, and pointing a browser at its root. We are greatly interested by these points and proposals, and we look forward to developing them in further technical exchanges.

5 Discussion

The translation industry needs an open nonproprietary format for packaging all of the materials necessary to complete a translation project, and the translation packet provides that support in direct compliance with industry standards [2, 3]. The translation packet format offers clear benefits for both buyers and providers of translation. An individual requesting a translation can create a packet without knowing beforehand who will perform the translation; instead, the specifications can guide the selection of the appropriate translator. Because all packets have the same structure and use the same parameters for the STSS, a translator will know exactly where to look for instructions rather than needing to search through correspondence. Translation tools can also use the packet to automatically load required translation memories and terminology resources without the need for the translator to search for them. The packet provides more than just materials for producing a translation; it provides a structure that promotes quality throughout the entire process, from authoring to publishing. The reliance on structured specifications allows the packet to go beyond the functionality of a single file format such as XLIFF.

We are currently developing a Web application that will function as a proof of concept for the translation packet format. A visitor to the site will be able to create a free user account to save personal STSS profiles, use an online specifications builder to create an STSS, and upload files to include in a translation packet. The Web application will then build a translation packet based on the STSS and uploaded files, which the user can thendownload from the site. The website is not a repository for files, but users will be able to save their own STSS profiles and optionally include them in a public library. The use of STSS profiles will facilitate the generation of specifications because a user may have multiple similar translation projects that need only minor changes in their specifications.

Future work in developing a standard packet for translation tasks includes finding a consensus between buyers and providers as to what the packet needs to accomplish, and then turning the specification over to an industry standards body such as OASIS [7] or ETSI [8] for standardization. Tool vendors will then be able to use import/export functions to read and create packets and even include proprietary files. However, we want to keep the format open enough that it will be of use to even those professionals who do not have a particular tool. Just as ISO intermodal shipping containers have helped to standardize the way goods are transported from point A to point B, the translation packet format will help to alleviate the need to manually organize and modify translation materials.

Sample translation packets will be made available at http://www.ttt.org/specs/ for inspection and comment. In addition, the proof of concept Web application for constructing translation packets will be linked via that site. We invite interested readers to send their comments and suggestions to larc.container@gmail.com.

6 References

1. Yewell, Smith. LISA 2011 Open Standards Summit, March 1, 2011,Danvers, MA.
2. ASTM F2575 06 Standard Guide for Quality Assurance in Translation, http://www.astm.org/Standards/F2575.htm.
3. ISO TC 37 WD 11669, http://www.iso.org/.
4. Zip Format Specification, http://www.info-zip.org/.
5. OpenPGP, http://www.openpgp.org/.
6. ISO 639-1, http://www.iso.org/.
7. ISO 3166, http://www.iso.org/.

8. Interoperability Now package outline: http://interoperability-now.org/tiki/tiki-index.php?page=The+Package.
9. MemoQ handoff/handback guide: TranslatingmemoQ handoff packages in other tools,http://kilgray.com/resource-center/user-guides(user authentication required).
10. M.T. Carrasco Benitez, Directorate-General for Translation of the European Commission, http://dragoman.org/med-template.html.