Linport as a standard for interoperability between translation systems

Alan K. Melby
Brigham Young University
Department of Linguistics & English Language
4064 JFSB
Provo, Utah, USA 84604 USA
akmtrg@byu.edu

Tyler A. Snow
Brigham Young University
Translation Research Group
4064 JFSB
Provo, Utah, USA 84604 USA
tylerasnow@gmail.com

Abstract

There is an increasing need for a vendor-independent standard regarding the packaging of translation-related projects to increase interoperability among translation tools worldwide. The Language Interoperability Portfolio (Linport) project, a result of collaboration among various organizations, presents a viable option for such a standard. Implementing Structured Translation Specifications (STS), Linport uses a design framework wherein projects can be documented in detail in a “portfolio” and then split out into discrete translation task packages, called "TIPPs", which can then be merged back into a revised portfolio after the tasks are completed. Linport will greatly enhance the cooperative efforts of myriad individuals involved in translation, serving to reduce costs, increase efficiency, and promote quality.

Introduction

The modern world sees increasing opportunities for connection and communication with others. Thanks to modern technologies such as satellites, cell phones, and the Internet, instant global communication is possible, a phenomenon unimaginable even one hundred years ago. This level of global interoperability, the capacity to work with others to accomplish tasks quickly and easily, is a defining achievement of our era. The great advancement of interoperability in the age of globalization will continue to make life easier. However increased interconnectivity does not necessarily bring interoperability. Specifically, it often requires the development of new standards to promote interoperability.

A classic example of interoperability breakdown comes from the shipping industry. For thousands of years countries and companies would ship products internationally. Eventually, there was a focus on determining means of cutting costs and travel time in order to maximize trade profits. During attempts to streamline these processes, it was realized that containers were being unloaded from cargo ships which then did not fit onto trains and boats that would then carry them to their final destinations. Further, there was great inconsistency among container types and sizes. Consequently, it was difficult for trucking or train companies to make appropriate plans for the movement of incoming goods. Containers would have to be unpacked, the goods removed from within, and then repacked into containers meeting the needs of the transportation company, wasting valuable time and money in the process. There was great need for a standard. Ultimately, a standard shipping container size was agreed upon by the international community. These standardized shipping containers could be easily moved from ship to truck without removing the contents as both the ship and the trucks were made to handle the exact dimensions of the pre-specified containers. These measures thus promoted enhanced interoperability on a global scale in the shipping industry.

The need for standards applies not only to the shipping industry but extends to many other fields as well, including translation technologies. There are currently many translation companies vying for their translation software to be used and recognized in the world of commercial translation. Some of these companies are SDL, LingoTek, MultiLing, Kilgray, SYSTRAN, and XTM-Intl, among many others. Each one of these companies has different interfaces that handle translation projects in different ways, oftentimes creating interoperability difficulties. For example, if one company starts a project using SDL tools and creates a project package, and then subcontracts out to a freelance translator who uses Kilgray, the same problem ensues as found in the shipping container example. The various project components, such as source text, specifications, and file types, all have to be "unpacked" from the SDL package and "re-packed" in the Kilgray-style package for the freelance translator to perform their allotted task within the project. Upon completion, it then has to be return to SDL format. Hence, similar to the shipping industry, the translation industry is in great need of a standard "container" of its own to package translation projects that allow for interoperability between the many translation software tools now available to professional translators and companies worldwide.

Linport: A Standardized Container for the Translation Industry

Fortunately, such a standardized container for the translation industry is already under development, known as the Language Interoperability Portfolio (abbreviated to *Linport*). It is intended to provide a complete and interoperable container solution for both translation projects and tasks within projects. In fact, portfolios will eventually be extended to include both authoring projects that have not yet been sent to a translation service and publication projects in which there is no designation of source language, such as certain official multilingual publications of the European Commission.

There is a need to clearly define what a translation project is. For the purposes of the present paper, we will use the implicit definition in ISO/TS 11669 (2012). A translation project has three phases: pre-production, production, and post-production. Typical tasks performed during the various phases are as follows:

- Pre-production
    - Development of project specifications
    - Preparation of the source content for translation, such as adaptation
    - Monolingual terminology research
    - Selection of a translation service provider

    - Pre-translation processing of source text against translation memories
- Production
    - Initial translation (by a human, MT, or a combination)
    - Self-checking by the human translator (or sometimes post-editing)
    - Revision (a bilingual task often performed by someone other than the person who did the initial translation)
    - Review by a subject matter expert (often by someone who does not know the source language)
    - Final formatting (or compilation in the case of software localization)
    - Proofreading
    - Additional tasks such as random quality control on sections of the translation or back translation
- Post-production
    - Assessment of the delivered translation product, the process used to obtain it, and the project as a whole, all relative to the project specifications

A Linport container, documenting the details of an entire translation project, will include information about the individual translation tasks that are to be performed during the project. Each of these tasks would be accessible to appropriate participants in the project who, upon task completion, would be able to “pack” their goods into a Linport container for further use. Keeping with the shipping example therefore, the Linport project would represent an overall project view much like a shipping container, but also would be able to define particular translation tasks within the project, comparable to standardized boxes that would be shipped in the larger container itself.

Elements of Linport

The data model for a Linport portfolio can be represented as an XML document instance containing two main sub-elements. The first is the portfolio information element (*portInfo*), similar to an HTML header element. It contains information about the portfolio as a whole, such as creator name, global specifications that apply to all tasks, and universal identifiers to facilitate breakdown and reintegration of the portfolio. The portInfo element also contains a table of contents listing the files that are part of the project. The second element is the *payload element*, containing the resources to be translated, as well as the supporting resources needed for initial translation or any other required task such as revision or review. Examples of these resources could include translation memory files, textual references, terminology files, style guides, among others.

As of this writing, the portfolio data model has not yet been formalized using a mechanism such as an XML schema definition language; however, an informal data model using XML as a skeleton is found in Appendix 1.

Details of the portfolio data model, such as names of elements, are subject to change and will be further discussed within the Linport community.

The other data model within Linport is for a task package, known as a TIPP. As of this writing, the TIPP data model is on version 1.5, and the details are available at:

www.interoperability-now.org

Together, the portfolio data model and the TIPP data model form the complete Linport data model.

Structured Translation Specifications (STS)

The ability to coordinate translation tasks and achieve quality is enhanced by the incorporation of Structured Translation Specifications (STS). The STS is found in Linport portfolios and also in Linport task packages. This will allow companies or translation project managers to specify important metadata about the translation itself, such as the target audience and intended use of the translation by providing consistent necessary information relative to translation tasks. An STS is developed by considering 21 important aspects of a translation project that should be considered even before entering the production phase of a translation project. Each aspect is called a translation parameter. The 21 parameters are provided in Table 1 and are available online at http://ttt.org/specs with additional explanations. Translation specifications are the values of these parameters for a particular translation project. The parameters remain the same while the specifications are particular to a project.

The 21 translation parameters used in the Linport project were not invented for Linport. They come from ISO/TS 11669, as does the definition of a translation project given above.

A. Linguistic [1–13]

- ***Source-content information [1–5]***
  - *[1] textual characteristics*
    - *a) source language*
    - *b) text type*
    - *c) audience*
    - *d) purpose*
  - *[2] specialized language*
    - *a) subject field*
    - *b) terminology*
  - *[3] volume*
  - *[4] complexity*
  - *[5] origin*
- ***Target-content requirements [6–13]***
  - *[6] target language information*
    - *a) target language*
    - *b) target terminology*
  - *[7] audience*
  - *[8] purpose*
  - *[9] content correspondence*
  - *[10] register*
  - *[11] file format*
  - *[12] style*
    - *a) style guide*
    - *b) style relevance*
  - *[13] layout*

B. Production tasks [14–15]

- *[14] typical production tasks*
    - *a) preparation*
    - *b) initial translation*
    - *c) in-process quality assurance*
- *[15] additional tasks*

C. Environment [16–18]

- *[16] technology*
- *[17] reference materials*
- *[18] workplace requirements*

D. Relationships [19–21]

- *[19] permissions*
    - *a) copyright*
    - *b) recognition*
    - *c) restrictions*
- *[20] submissions*
    - *a) qualifications*
    - *b) deliverables*
    - *c) delivery deadline*
- *[21] expectations*
    - *a) compensation*
    - *b) communication*

Table 1. Standardized Translation Project Parameters.

History of Linport

Cooperative work to date

Linport itself is an example of a conglomeration of companies and organizations already working together. The project comes from three main project streams. In March 2011 many of the organizations that participated in the former LISA standards organization agreed that a container type format was needed in the translation industry. The Globalization and Localization Association (GALA) and the Localization/Translation and Authoring Consortium (LTAC) [www.ltacglobal.org] began work on what was named the Container project (Melby et al., 2012). The first presentation of their work was given a month later in Torino, Italy at the JIAMCATT translation technology conference (JIAMCATT). After the presentation, a representative of the Directorate General of Translation of the European Commission (EC) indicated that their organization already had started on a similar project, known as the Multilingual Electronic Dossier (MED) project. In MED, they aimed to represent an entire translation project in what they called a translation "dossier" (Melby et al., 2012). After a series of discussions, the Container project and the MED were merged to form the Linport project in July 2011, hosted by the non-profit organization LTAC Global. It was decided that the Linport container for an entire project, be it authoring, translation, or publication, would be called a portfolio.

Work by Interoperability Now!

An initiative company called Interoperability Now!, or IN!, had already started work on yet another similar project in 2010. After the participants in the Linport project and those involved in the IN! project gained awareness of one another's work and held a series of discussions, IN! agreed to integrate their "container" format for a translation-related task into the Linport project

(Melby et al., 2012). This final step in the merger of the three streams occurred around June 2012.

Additional Support

Once the three streams (The LISA Container Project, the EC MED Project, and the Interoperability Now! TIPP project) had agreed to work under the Linport umbrella and develop a single standard, to avoid competing standards that accomplish much the same thing, translation tool vendors and translation companies were approached for support. Several tool vendors and translation companies have expressed explicit support (http://www.linport.org/suporg.php). Without the support of these organizations, and in particular, the Gold Level sponsor, Multiling, Linport would have been able to continue.

IN!'s Translation Interoperability Protocol Package (TIPP)

The primary and essential contribution to the Linport project by IN! is the Translation Interoperability Protocol Package (TIPP). TIPP represents a single translation task to be performed using exactly two languages to be used in a translation workflow, instead of representing the whole translation project, which could potentially involve many languages and tasks. By design then, a Linport portfolio should theoretically be able to be broken down into multiple TIPP task packages (TIPPs) which can be accessed, completed, and reintegrated back into the portfolio for transportation to another translation tool (Melby et al. 2012).

The Viability of Linport

So far in this paper, we believe that we have established the need for standard containers, one for all the elements of a translation project and the other for the information needed to perform one task within a project, and we have presented the Linport project's approach to addressing this need (namely, a portfolio for a project and a TIPP package for a task). How viable is the Linport project? That is, how likely is it that Linport will result in a standard that is widely implemented by stakeholders in the translation industry, including content owners, translation technology developers, and translation companies?

Extensive discussions with translation tool developers during the fourth quarter of 2011 revealed that they were adamant on one point: they would not be willing to implement any kind of container standard unless all three players (the LISA effort, taken over by GALA, the European Commission effort, and the Interoperability Now! Effort) joined together to define one standard rather than competing with each other and trying to get the tool vendors to choose sides. As mentioned above, this hurdle has been overcome, so long as the Linport coalition holds.

An additional hurdle was the obvious need for the system for developing and representing translation project specifications to be compatible between the representation of a project (a portfolio) and the representation of a particular project task (a TIPP package). Fortunately, the three streams forming Linport have agreed to use exactly the same system for project parameters: ISO/TS 11669, which has published in May 2012. During the merger discussions between June 2011 and June 2012, ISO/TS 11669 was already a draft international standard, was known the parties, and was agreed on, pending its publication.

Once ISO/TS 11669 was published, it became urgent to establish that it is practical to use it to create a machine-readable set of structured translation specifications. In early 2012, The BYU (Brigham Young University) TRG (Translation Research Group) began work on an open-source software tool that allows an STS (ISO/TS-11669-based structured translation specification object) to be built in a manner that is in no way tied to a particular translation tool.

This open-source software package has matured to the point where it has entered beta testing by a few members of the Linport community. As of this writing, documentation is still lacking, but the on-line STS building tool is available at:

http://app.linport.org/

The STS builder, which also builds a TIPP independent of any particular translation tool, was demonstrated on September 20, 2012, during a webinar hosted by GALA.

A recording of that webinar is available at:

http://tinyurl.com/LinportWebinarSep2012

or

http://www.gala-global.org/linport-project-update

The STS and TIPP builder is written in Django for rapid development but could easily be ported to another software development platform and integrated into a commercial tool if desired.

The use of a TIPP to allow a translation task to be requested, performed, and a response TIPP sent back to the requesting tool was demonstrated at the June 2012 Localization World conference in Paris. The slides for this demonstration are available at:

http://code.google.com/p/interoperability-now/wiki/conferences

The current question concerning the viability of the Linport project is whether the Linport portfolio data model and the Linport task-package (TIPP) data model are compatible.

We define portfolio-TIPP compatibility by success in satisfying the following three criteria:

1. A Linport portfolio can adequately represent an overall translation or project;
2. A portfolio can be broken down into individual translation tasks(represented in TIPP format), and
3. A Linport portfolio can be reassembled as a whole following the completion of individual translation tasks.

Given the entirely separate origins of the portfolio data model (i.e. the European Commission MED project) and the TIPP data model (i.e. the Interoperability Now! initiative that was industry, not government based and was not even aware of the MED project until TIPP was quite far along, it is not obvious that the two data models are compatible.

If Linport can adequately represent a translation project and be broken down into individual tasks to the extent that each phase of the project can effectively be identified and accomplished and then reintegrated back into a portfolio, then as it stands Linport is a very promising representation that can viably be used in all translation tools. If this is not the case however, then Linport is not ready for use in the translation industry and will require further design modification and testing.

For our study, a sample Linport portfolio was created using the information from a real-world translation project donated by the translation company MultiLing, Inc. Each of the steps requisite to demonstrating the viability of a Linport portfolio was carefully followed.

Parsing and Merging Software

Software to perform the parsing of a portfolio into TIPPs and consequent merging of the completed tasks back into the portfolio was created and presented in an October 2012 Linport conference call. The program uses an XML tree parser extract from the portfolio the information needed to form the TIPPs, and then uses the same modules and methods in reverse to scan the TIPPs to merge back into the portfolio after they have been completed. The split and merge application is written in Perl. It is short and simple to allow future programmers to port the program into other programming languages.

MultiLing Sample Portfolio

The first Linport portfolio used to test the program was created by hand using the portfolio skeleton found at http://dragoman.org/linport/ldm.xml. The representation of the data itself was very effective. The Linport Data Model (LDM) was able to represent all desired information about the project within the XML format. (See the portfolio in the appendix for the results). The source text and all files associated with the translation were represented in a portfolio based on the LDM skeleton.

Integration of the TIPP back into the portfolio went as smoothly as could be reasonably anticipated. It was found that there were two ways to merge the TIPP into a portfolio. The first way involved creating a blank portfolio filled in solely by the data contained in the TIPP. This was possible and there was no loss of information in the merge. This however was based on one very important condition: the TIPP must have included in its non-manifest data an STS XML file in order to recreate an STS in the portfolio. If the STS file was not included in the TIPP, then the STS would just be blank in the portfolio. While this situation would not destroy the portfolio on a whole, the portfolio would certainly be far less functional without this information. Actual results from the split-merge experiment are in Appendices 1 and 2.

Further discussion within the Linport community is needed on the question of when an STS is optional and when it is required.

The other way to integrate the TIPP back into the portfolio was to use an existing portfolio with other data already stored there and then add in the targeted TIPP data. This was a highly successfully measure, yielding no data loss. The TIPP information was able to be stored in the payload of the Linport portfolio. There were some minor STS issues that were resolved by making sure that the new STS that went with the re-integrated document only used self-contained data to differentiate it from the rest of the portfolio which would follow the directives of the generalized portInfo directives. To use our earlier shipping analogy, the use of self-contained data in the STS would be similar to placing a box of oranges into a shipping container carrying mostly metal toy parts. The next time that container is unloaded the portInfo section would treat all the metal parts as with general instructions but read the oranges box differently and handle them appropriately.

Conclusion

In summary, Linport was able to successfully model an entire translation project using a sample real-world translation project from MultiLing.. It was successfully split into individual translation tasks, represented in the form of TIPPs, and merged back together into a complete portfolio with no loss of data. This implies that Linport is ready for further testing. Time will tell whether tool vendors will start to implement them when designing translation tools. If they choose to do so then the entire industry would benefit from the ease of transition between tools, saving valuable time and money.

The Linport work in 2012 has been based on a high-level design consisting of principles, assumptions, and requirements agreed on by the Linport community in late 2011. The high-level design of Linport has survived to date. See Appendix 3 for the Linport design.

Future Work

There is much additional work to be done with Linport.

The next step is to formalize the portfolio data model, using an XML schema definition language and other methods. Validation tools must be developed, and then many additional sets of translation project data, some very complex, must be tested with the portfolio model, splitting them into TIPPs, performing the requested tasks, and combining the response TIPPs back into a revised portfolio.

Before much additional testing can be done, an open-source tool will be needed to facilitate the creation of portfolios.

So far, one other project has been received, a project provided by the DGT (Directorate General of Translation), and it will be tested by the time this paper is presented at the ASLIB TC conference in November 2012.

Additionally, some early adopter translation tool developers and content owners will need to try out the Linport portfolio, considered by the Linport community at this stage to be a “blueprint.” as presented here in this paper. A blueprint, as used here, is the basis for a standard that has not yet been submitted to a standards body.

Some early adopter tool developers will implement the TIPP format; others (particularly workflow management tools) will also implement the portfolio format.

As more experimentation is done, flaws in the design of the portfolio and the formats will be discovered and addressed. There is no one absolutely correct way to define the project and task containers. The key requirements are keeping the Linport coalition from splintering and demonstrating viability of whatever the Linport community agrees on.

Linport is not a standards body. At some point, the Linport community will decide that the Linport blueprint is sufficiently mature to be submitted to an industry standards body.

Due to the increases in efficiency and interoperability that Linport will help to promote, it is undoubted that numerous parties will benefit from the work of this project, including content owners, tools developers, translation companies, translators, translator trainers, and machine translation developers. The Linport community anticipates presenting the Linport data models and accompanying design to the OASIS Standards Committee (or the ETSI LIS ISG) sometime in 2013. Once Linport becomes an industry standard (Oasis or ETSI), Linport will be able to take an accelerated course to becoming an ISO standard within Technical Committee 37. Linport will then be recognized by the same international organization that standardized containers in the shipping industry. Both Oasis and ETSI have formal relationships with ISO.

The most important future work item for Linport is to promote implementation by tool vendors and use by all stakeholders.

References

2012. The International Annual Meeting on Computer-Assisted Translation and Terminology (JIAMCATT) website. Retrieved from http://jiamcatt.org/.

2012. Linport: The Language Interoperability Portfolio Project. Retrieved from http://linport.org.

2012. Structured Specifications and Translation Parameters. Retrieved from www.ttt.org/specs.

ISO/TS 11669: Translation projects. General guidance. 2012. ISO (www.iso.org)

Melby, A., Lommel, A., Rasmussen, N., & Housley, J. 2012. “The Language Interoperability Portfolio (Linport) Project: Towards an Open, Nonproprietary Format for Packaging Translation Materials.” (*Unpublished article submitted to JIAL).*

Melby, Alan. Chandler, Brian. Lommel, Arle. 2012. “Linport addresses translation package compatibility.” *Multilingual Magazine*. July/Aug: 45-47.

Melby, Alan. 2011. “The Seoul of Standards and You.” *The ATA Chronicle*. Oct: 12-16.

Appendix 1

The Linport Portfolio Data Model as of October 2012

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<!--
 XML representation of the Linport[Portfolio] Data Model - draft 9 -
10 Oct 2012
```

```
 Directory structure representation elsewhere
-->

<portfolio>
 <portinfo> <!-- "portinfo" similar to HTML "header" -->
  <metadata> <!-- the elements inside metadata are specified in the
Linport Metadata Blueprint -->
   <sts></sts>
   <tasktype>footask</tasktype>
   <identifier></identifier>
   <creator>M.T. Carrasco Benitez</creator>
   <trace></trace>
   <foometa>http://dragoman.org/linport/metadata.html</foometa>
  </metadata>
  <list>
   <global></global>
   <toc></toc>
   <hashtype>md5</hashtype>
  </list>
 </portinfo>
 <payload> <!-- "payload" similar to HTML "body" - "payload" can
contain N "grouplang" -->
  <grouplang xml:lang="en"> <!-- source - "grouplang" can contain N
"docplus" -->
   <docplus name="foodoc" format="xml">
    <doc uri="http://example.com/foo" />
    <aux> <!-- "aux" can contain N elements -->
     <metadata>
     </metadata>
    </aux>
   </docplus>
  </grouplang>
  <grouplang xml:lang="es">
   <docplus name="foodoc" format="xml">
    <doc></doc>
    <aux>
     <metadata>
     </metadata>
    </aux>
   </docplus>
  </grouplang>
  <grouplang xml:lang="fr">
   <docplus name="foodoc" format="xml">
    <doc></doc>
    <aux>
     <metadata>
     </metadata>
    </aux>
   </docplus>
  </grouplang>
 </payload>
</portfolio>
```

A Sample STS used in both portfolios and TIPPs. Parent-child STSs are allowed in portfolios.

```
        <structuredTranslationSpecifications type="child" id="4107f199-1745-4f4e-
9481-cfac8505fda7">
            <source>
                <textualCharacteristics status="inherited">
                    <language></language>
                    <textType></textType>
                    <audience></audience>
                    <purpose></purpose>
                    </textualCharacteristics>
                    <specializedLanguage status="inherited">
                    <subjectField></subjectField>
                    <terminology></terminology>
                    </specializedLanguage>
        <volume>status="inherited">
                    </volume>
                    <complexity status="inherited"></complexity>
                    <origin status="inherited"></origin>
            </source>
            <target>
                <targetLanguageInformation status="inherited">
                    <language></language>
                    <targetTerminology>
                    </targetTerminology>

    </targetLanguageInformation>
            <audience status="inherited"></audience>
            <purpose status="inherited"></purpose>
            <contentCorrespondence status="inherited"></contentCorrespondence>
            <register status="inherited"></register>
            <fileFormat status="inherited"></fileFormat>
            <style status="inherited">
                <styleGuide>.</styleGuide>
                <styleRelevance></styleRelevance>
            </style>
            <layout status="inherited"></layout>
        </target>
        <production>
            <typicalTasks status="inherited">
                <preparation></preparation>
                <initialTranslation></initialTranslation>
                <qualityAssurance></qualityAssurance>
            </typicalTasks>
            <additionalTasksstatus="inherited"></additionalTasks>
        </production>
        <environment>
            <technology status="inherited"></technology>
            <referenceMaterials status="inherited" path="internalFile">
            </referenceMaterials>
            <workplaceRequirementsstatus="inherited"></workplaceRequirements>
        </environment>
        <relationships>
            <permissions status="inherited">
                <copyright></copyright>
                <recognition></recognition>
                <restrictions></restrictions>
            </permissions>
            <submissions status="inherited">
                <qualifications></qualifications>
                <deliverables></deliverables>
                <delivery></delivery>
                <deadline></deadline>
            </submissions>
            <expectations status="inherited">
                <compensation></compensation>
                <communication></communication>
            </expectations>
        </relationships>
    </structuredTranslationSpecifications>
</metadata>
</aux>
</docplus>
</grouplang>
</payload>
</portfolio>
```

Appendix 2

Translation Interoperability Protocol Package (TIPP) Example

A TIPP consists of two parts:
- The manifest, an XML file defined by the TIPP Schema.
- An (optionally encrypted) Package Object Container. The Package Object Container is a ZIP archive containing one or more Package Objects. The Package Object Container is also referred to as the Payload.

These two parts are contained in the package itself, which is also called the Envelope. In the current implementation, the Envelope is a ZIP archive.

There is no folder structure within the Envelope. The manifest must be named manifest.xml. The Package Object Container which must be named PackageObjects.zip.enc if encrypted or PackageObjects.zip, if unencrypted.

An Example of a TIPP XML manifest created from this portfolio using the software mentioned:

```
<TIPPManifest version="1.4">
<GlobalDescriptor>
<UniquePackageID>urn:uuid:4107f199-1745-4f4e-9481-cfac8505fda7</UniquePackageID>
<PackageCreator>
<Name>Tyler Snow</Name>
<ID>http://example.com/</ID>
<Update>2012-11-07T01:26:04Z</Update>
<Tool>
<ToolName>Linport Splicer</ToolName>
<ToolID>None</ToolID>
<ToolVersion>1.0</ToolVersion>
</Tool>
</PackageCreator>
<TaskRequest>
<Task>
<TaskType>http://schema.interoperability-now.org/tipp/v1.4/tasks/translate-strict-bitext
</TaskType>
<SourceLanguage>es-ES</SourceLanguage>
<TargetLanguage>en-us</TargetLanguage>
</Task>
        </TaskRequest>
    </GlobalDescriptor>
    <PackageObjects>
        <PackageObjectSection type="http://schema.interoperability-
now.org/tipp/v1.4/tasks/translate-strict-bitext/sts" name="sts">
        <ObjectFile sequence="1">
            <Location>El Eclipse-sts.xml</Location>
        </ObjectFile>
        </PackageObjectSection>
    </PackageObjects>
    </TIPPManifest>
```

Appendix 3

Portfolio specifications found at: http://www.linport.org/design.php :

Linport: Design Requirements, Principles, and Assumptions

Last updated: November 28, 2011

This page outlines the design requirements, principles, and assumptions of Linport. The contents of this page are subject to revision and discussion.

Requirements

The following are proposed requirements for the output of the Linport project:

Technical

- **Self-contained translation project information**. The package must contain or point to all information needed to carry out tasks within translation projects.
- **Feasibility of implementation**. It must be possible to develop import/export routines for existing translation tools without requiring substantial modification to their basic design.
- **Platform/translation tool independence**. Linport must not be tied to any specific platform, translation tool, or programming language.
- **Structured Translation Specifications**. Linport must include a way to convey specifications compatible with the forthcoming [published May 2012] ISO/TS 11669 (see www.ttt.org/specs/).
- **Support for remote and local access to resources**. Resources referenced in a Linport instance may be local (e.g. on a hard drive) or remote (say on a server, accessed via Web services) and both modes of access must be supported.
- **Open interface**. An abstract application software interface description, and several programming-language-specific APIs (all derived from the abstract interface description) and a library for each API, so that applications can use the library rather than accessing a Linport package directly.
- **Reference implementation**. It is insufficient to define just a format or interface. A reference open-source implementation of Linport must be provided.
- **Package modularization (splitting and joining)**. Packages must support joining and splitting operations, as needed to support workflow requirements. E.g., a complex multilingual package might be split into multiple bilingual packages, which could then be modified (e.g., through adding a target text) and recombined later on.
- **Profiles**. Linport must allow for profiles, subsets of the overall Linport architecture that support the needs of specific audiences or purposes.

Collaborative

- **Integration with TIPP**. The proposed solution must be compatible with TIPP from Interoperability Now!. We do not believe that developers will implement two solutions to accomplish essentially the same task.

Principles

These principles guide the development of the Linport format. The Linport Project will:

1. Encourage translation/localization tool vendors and content owners/producers to be early implementers and provide feedback.
2. Strive for simplicity (so the format is easy for tool vendors and content owners to implement).
3. The scope of the Linport project does not include the development of payload formats, but it may involve specifying preapproved standard payload formats in strict versions of Linport. The Linport development process, however, does include cooperation with groups that are developing payload formats (e.g., XLIFF).
4. The Linport format should be based in part on existing tool- or organization-specific package formats.
5. The initial focus will be on profiles for translation/localization, but allow for profiles to support content management systems and authoring processes in the future.

Assumptions

The following assumptions inform the work of the Linport project:

1. 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.
2. The documents describing the Linport format should be available as a free download with a Creative Commons Attribution 3.0 license. Any reference implementation software should freely available, including source code, royalty free, and with multiple open source licenses to choose from.
3. Competition among package formats from multiple projects should be avoided.