

Introduction

As we see the ever increasing adoption of XML into the publishing domain, it becomes more obvious that certain things are missing from the standard perspective. As usual, proprietary solutions quickly appeared to plug the gaps but with the commensurate draw backs: lack of openness and transparency.

One of the most fundamental aspects of Open Standards is how well designed they are. It is not surprising: take a group of involved industry experts, a democratic charter, and peer and public review, and you usually end up with a well-designed solution. I never cease to be surprised how much better Open Standards-based solutions are than their proprietary equivalents. No wonder many people refer to RTF as "really terrible format."

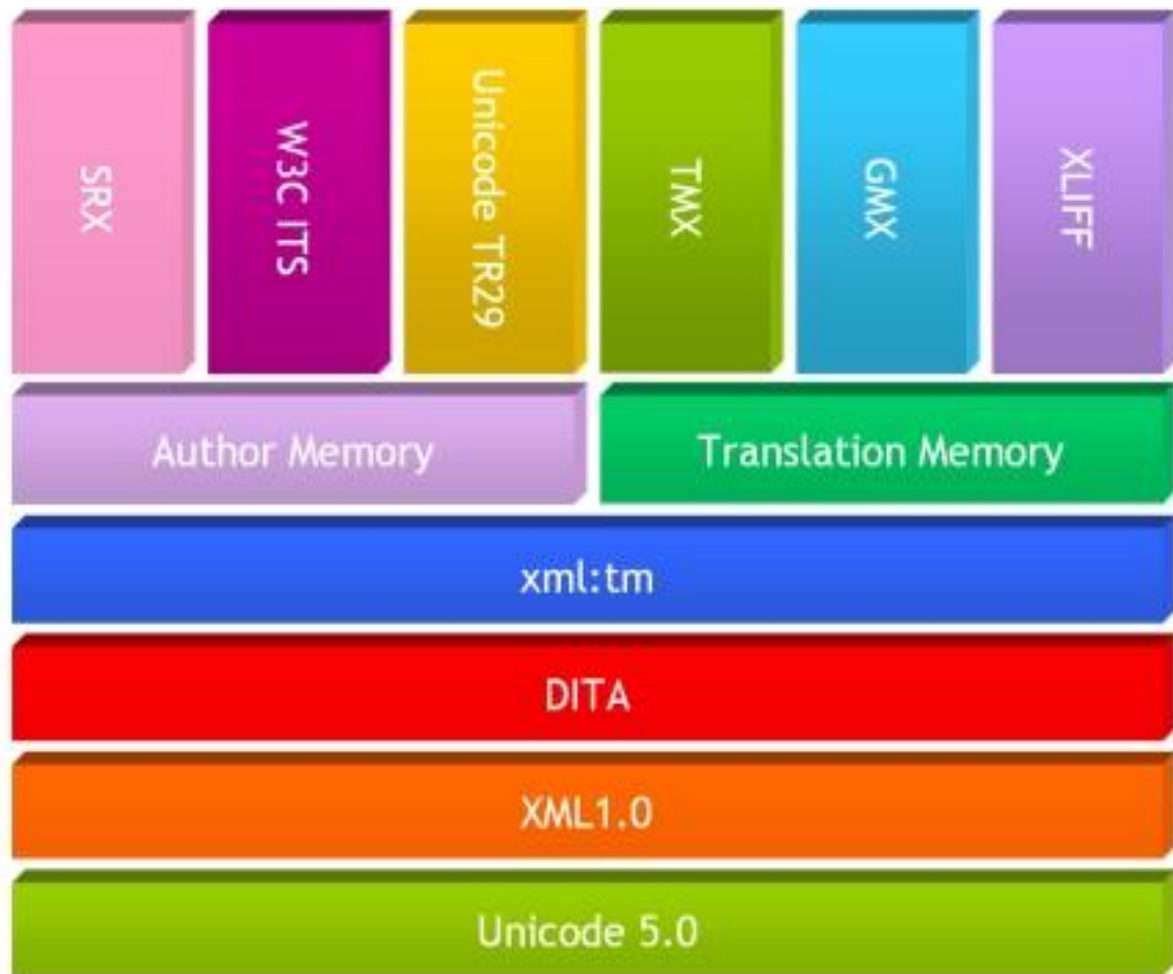
Open Standards also provide an example of IT best practice: You create a specification by talking to all of the interested parties, publish the results for public comment, and then print the results. Yes, rarely things do not go according to plan, but the nature of standards allows for revision and review in light of practical feedback. Rather like democracy, standards have the ability for self correction.

OAXAL stands for Open Architecture for XML Authoring and Localization and is a newly founded OASIS reference architecture technical committee. It covers all of the aspects of technical publishing to create an open and effective solution.

OAXAL—the basics

One of the things that XML kicked off was a veritable explosion of Open Standards. The main reason is that XML provided the necessary extensible vocabulary. The other reason has been the dramatic reduction in communication costs that allows for cheap and regular teleconferencing throughout the world.

OAXAL is made up of a number of core standards from W3C, OASIS, and, last but not least, LISA, the Localization Industry Standards Association. The following is a diagrammatic representation of the OAXAL standards component stack:



Let us have a look at all of these standards in detail:

Unicode

Those of us who remember the "bad old days" before Unicode, praise it every day. The "tower of Babel" (of various illogical and contradictory encoding schemes that preceded Unicode) was the cause of much grief to anyone involved in the translation of electronic documents.

XML

Where would we be without XML? It has been a monumental standard that has given us the extensible language that was lacking previously. It was as if, finally, the IT industry was given a common language to talk to one another. It is not perfect, but rather like democracy...all of the alternatives are so much worse that anything else is not worth considering. Coming on the back of the lessons learned from SGML, XML will remain for many years the fundamental building block of all sensible IT systems. Interestingly enough, the adoption of XML in the publishing industry has been much slower than in computer science in general. There have been many reasons for the delays, but with Open Office and the latest version of Microsoft Office, the final hurdles have been breached.

A few things that are not that well appreciated are the following:

- You should always use UTF8 or UTF16 encoding for XML documents, monolingual documents as well as those destined for translation Unicode provides the basic typographical elements that most documents require.
- You should always use the xml:lang attribute on the root element, and anywhere within a document that a change of language occurs, to denote the language of the document.

W3C ITS

ITS stands for Internationalization Tag Set. It is the contribution from W3C to the localization process: <http://www.w3.org/TR/its/>

ITS allows for the declaration of Document Rules for localization. In effect, it provides a vocabulary that allows the declaration of the following for a given XML document type, for instance DITA:

- Which attributes are translatable?
- Which elements are "in line" that is they do not break the linguistic flow of text, such as "emphasis" elements?
- Which inline elements are "sub flows"? That is although they are inline, they do not form part of the linguistic flow of the encompassing text. For instance, "footnote" and "index" markers are inline elements.

W3C ITS provides much more, including a namespace vocabulary that allows for fine tuning localization for individual instances of elements within a document instance. W3C ITS, is therefore, at the core of localization processing.

Standard XML vocabularies

DITA, DocBook, XHTML, SVG – all of these standards dramatically reduce the cost of XML adoption. One of the factors that initially limited the adoption of XML was the high cost of implementation. XML DTD and Schema definitions are neither simple nor cheap. The key benefit for having standard XML vocabularies is that they reduce the cost of adoption. Additionally, tools and utilities arrive on the scene that lower the adoption price, sometimes dramatically. Not only so, but, as is the case with DITA, they often introduce key advances in the way we understand, build, and use electronic documentation.

xml:tm

xml:tm is a key standard from LISA OSCAR: <http://www.lisa.org/XML-Text-Memory-xml.107.0.html>

Think of xml:tm as the standard for tracking changes in a document. It allocates a unique identifier to each translatable sentence or standalone piece of text in an XML document. It is a core element of OAXAL because it links all of the other standards into an elegant, integrated system.

At the core of xml:tm are the following concepts which together make up 'Text Memory':

- Author Memory
- Translation Memory

You can think of Author Memory in terms of change tracking but also as a way to ensure authoring consistency – a key concept in improving authoring quality and reducing translation costs.

As far as Translation Memory (TM) is concerned, xml:tm introduces a revolutionary approach to document localization. It is very rare that a standard introduces such a fundamental change to an industry. Rather than separating memory from the document by storing all TM data away from the document in a relational database, xml:tm uses the document as the main repository with no duplication of data. This approach recognizes, fundamentally, that documents have a life cycle and that within that life cycle, documents evolve and change. At regular stages in that cycle documents require translation.

SRX

SRX (Segmentation Rules eXchange) is the LISA OSCAR standard for defining and exchanging segmentation rules: <http://www.lisa.org/standards/srx>

SRX uses an XML vocabulary to define the segmentation rules for a given language and to specify all of the exceptions. Segmentation refers to the process of dividing a document into translation-ready parts such as paragraphs or list items. SRX uses Unicode regular expressions to achieve segmentation. The key benefit of SRX is not so much exchange as it is the ability to create industry-wide repositories for the segmentation rules for each language. To this end, companies such as Heartsome, Max Programs, and XML-INTL have donated their own rule sets to LISA.

Unicode Technical Report 29

Unicode does not end with the encoding of character sets. The technical reports that form part of the standard and are included as an annex are equally important. TR 29 stands out as the way to define what constitutes words, characters, and punctuation. If you are writing a tokenizer for text, Unicode TR29 is where you start: <http://www.unicode.org/reports/tr29/>

Tokens are small bits of text that can be placed into larger documents via simple placeholders, like %site-name or [user].

TMX

Translation Memory eXchange is the original standard from LISA: <http://www.lisa.org/standards/tmx>

TMX helped break the monopoly that proprietary systems had over translation memory content. TMX allows customers to change systems and Language Service Providers without losing their TM assets.

GMX

Global Information Management Metrics Exchange is a three-part standard from LISA concerning translation metrics. GMX/Volume treats the issue of what constitutes word and character counts as well as allowing for the exchange of metric information within an XML vocabulary: <http://www.lisa.org/standards/gmx>

Believe it or not, before GMX/V there was no standard for word and character counts. GMX/V defines a canonical form for counting words and characters in a transparent and unambiguous way. The two associated standards will be GMX/C for 'complexity' and GMX/Q for 'quality'. These are still to be defined. Once the three GMX standards are available, they will provide a comprehensive way of defining a given localization task.

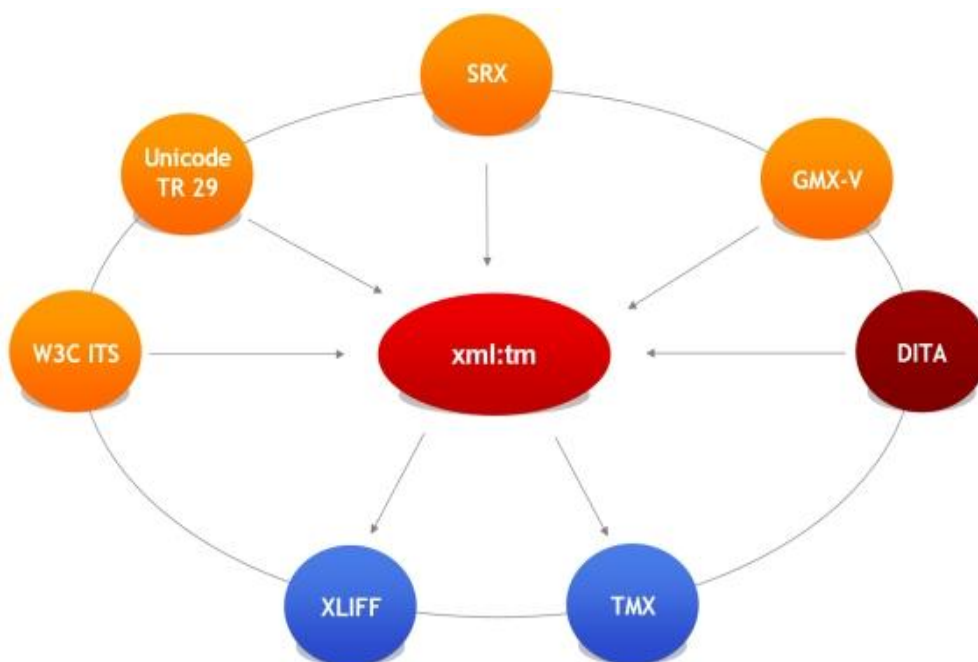
XLIFF

XML Localization Interchange File Format is an OASIS standard for the exchange of data for translation: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

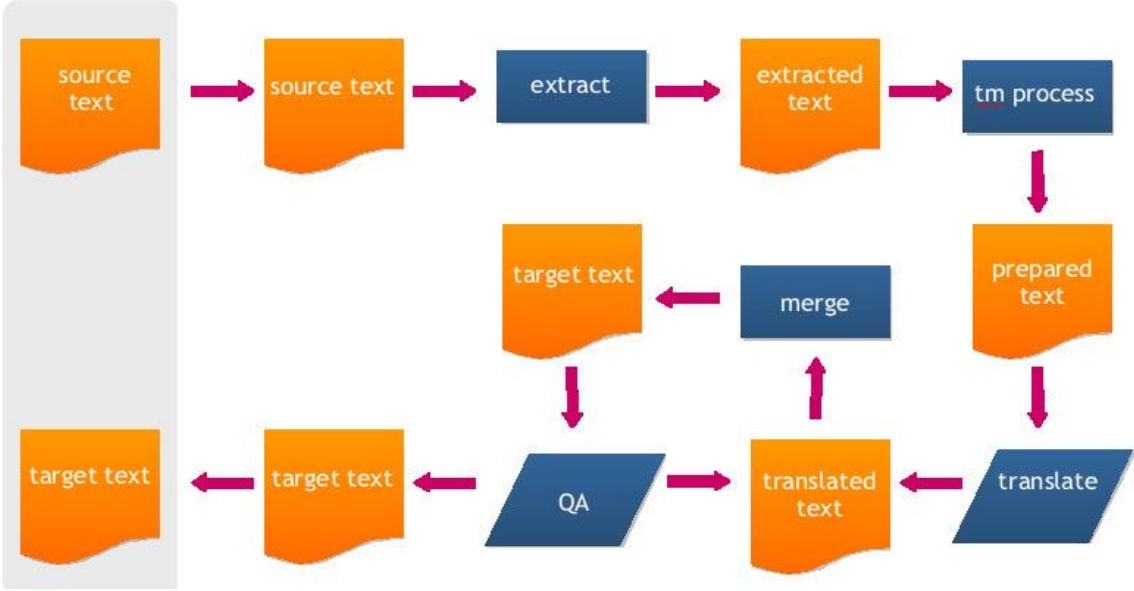
Rather than having to send full unprotected electronic documents for localization, with the inevitable problems of data and file corruption, XLIFF provides a loss-less way of round tripping text to be translated. Localization Service Providers, rather than having to acquire/write filters for different file formats or XML vocabularies, have merely to process XLIFF files, which can include translation memory matching, terminology, etc. Similarly, Computer Assisted Tool (CAT) providers have only the one format to deal with, rather than a spectrum of different original or proprietary exchange formats.

Putting it all together

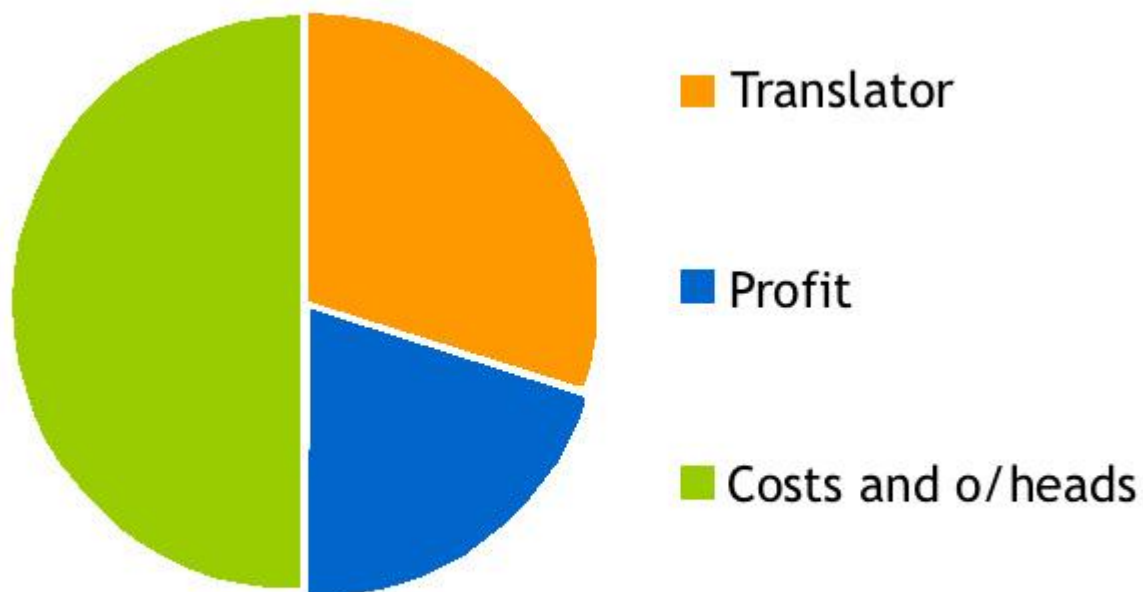
All of the above mentioned standards can be put together in the following elegant architecture:



Why does this matter? The answer is simple but not necessarily obvious at first glance. Prior to OAXAL, the typical workflow for a localization task looked as follows:



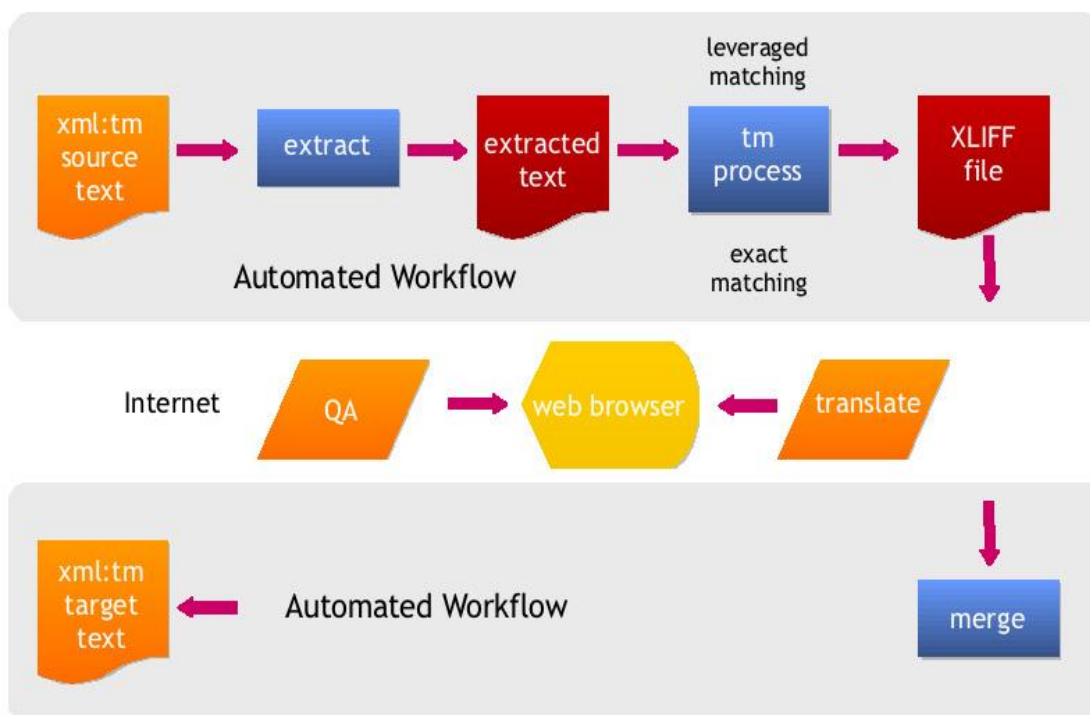
This workflow illustrates how the vast majority of localization tasks are conducted. Each arrow is a potential point of failure, as well as being very labor intensive. At the ASLIB conference in 2002, Professor Reinhard Schäler of the Localization Research Center at Limerick University presented the standard model for the Localization Industry:



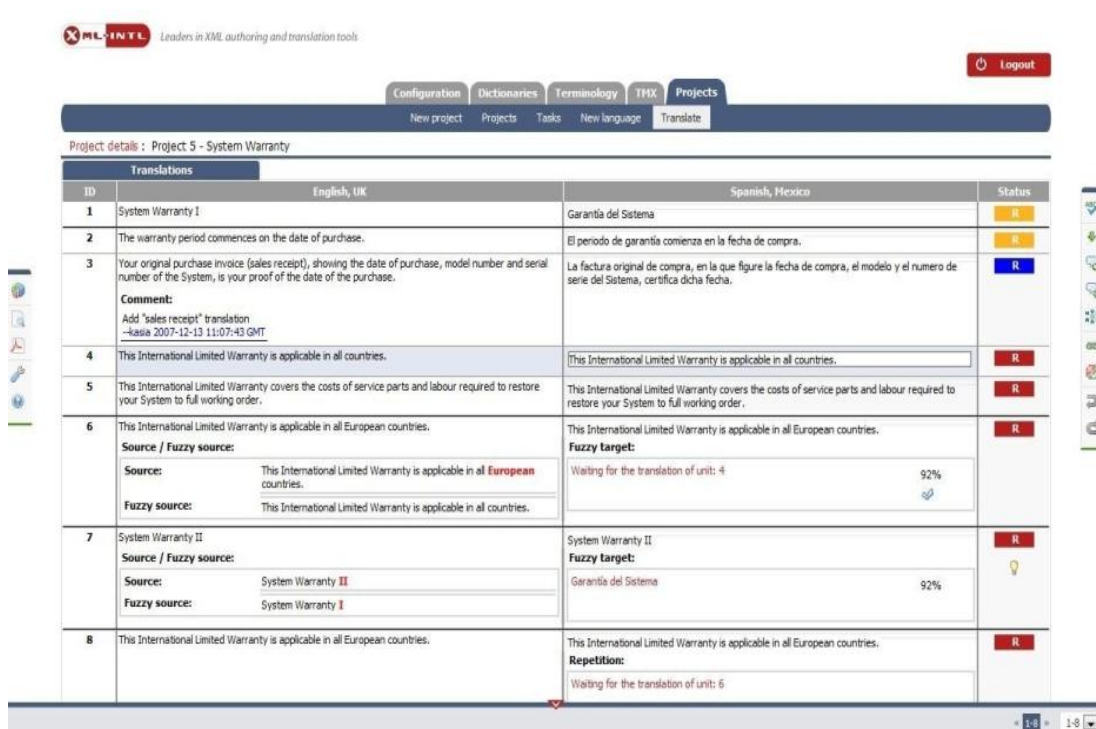
Source Professor Reinhard Schäler LRC - ASLIB 2002

Over half of the cost of a localization task is consumed in project management costs, which is a very error prone and labor intensive way of doing things.

With OAXAL, you can automate the complete workflow as follows:



This workflow provides considerable cost savings and improves speed, efficiency, and consistency. It also allows for a standard and consistent way of presenting the text to be translated via a browser interface, which further removes many manual processes. The current generation of Web 2.0 browsers allows for the creation of a fully functional translator workbench via a web browser, including the ability to have multiple translators working on the same file, auto propagation of matches within the file being worked upon, as well as support for infinitely large files.



Translators' work is constantly saved as well as written to a translation memory database for immediate availability.

The Proof

So much for the theory. None of these ideas would be convincing without a reference implementation. Otherwise this discussion would be no more than an interesting academic exercise. Thankfully, OAXAL is backed up with a real life successful proof of concept: www.DocZone.com. DocZone is a SaaS solution for technical documentation. It is a comprehensive, web-based XML publishing solution that encompasses all aspects of XML authoring and localization and a full implementation of OAXAL. DocZone comprises an XML content management system, XML editor, the XTM suite of author memory, and CAT tools from XML-INTL. In addition, it breaks the mold of expensive XML publishing systems and provides a subscription or pay per use model to remove the usual inhibitors to implementation of such systems. Up to now it would have been difficult to persuade people that an economic XML publishing solution would be available. Thanks to open standards within an OAXAL framework, DocZone has achieved such a solution.