XML The New Bandwagon

by René Snel

What Do We Do With XML?

How is XML shaping the localization process? This article looks at how electronic publishing concepts are changing from a linear approach to a highly interactive approach and how this will impact localizers. The article also looks at how translation tools will embrace XML through intelligent use of meta-data. This will be illustrated with some examples.

A Bit of History

First there are propriety source documents like FrameMaker, QuarkXpress, Word, Interleaf, etc. These documents are linear documents (from start to finish) and contain numerous formatting information. With the aid of additional tools, content is separated from formatting information; thus translators can concentrate purely on content. Both content and formatting information are stored for future leverage and reuse. On completion of translation, content and formatting are merged again for desktop publishing, book building, and in-context linguistic review.

The good news is that markup languages like SGML and XML separate translatable content from formatting. You may think that it should make life easier for the translators. Surely, the translator can concentrate on translation without having to deal with the formatting issues. Unfortunately, this is not exactly the case. In addition to linguistic skills, translators are expected to have some basic knowledge of HTML, software strings, scripts, and generally the behavior of interchangeable text formats like RTF, MIF, and Interleaf ASCII. Just as the nonlinguistic issues have come under control, now the demand to understand XML and the many XML flavors is even greater. Translators will be faced with out-of-context data, which is hard to display, translate, and proof.

Structured Documentation

A structured document generally allows you to break content into discrete content elements (fragments). These elements are placed and structured according to a particular rule or definition (Document Type Definition or Schema). For example, a *title* must be the first element in a *chapter* element, followed by an *introduction* element, followed by a *subtitle*, etc. By structuring a document this way, many applications can read and understand what type of document it is without reading the text. From this example we know that this is a chapter and must be treated accordingly. An element may have intelligence "behind the scenes." This intelligence is inserted by using meta-data or attributes. You may decide to translate the entire chapter except the title. You can indicate this by using meta-data: `<title translation="no">`

Because of this flexibility and diversity, many publishers are changing from FrameMaker to FrameMaker+SGML, to ArborText, from HTML to XML. In essence, this means a change from unstructured to structured documentation. To emphasize this trend: Adobe's hugely popular PDF format has now incorporated structural information and meta-data in the latest release. Historically, PDF has been a medium to deliver documents with a page-orientated format and is now adding structural information. Translation tools, including translation memory systems, will most likely follow this trend. Unlike SGML, XML makes it easy to understand structure. The ability to understand structure will be a basic requirement.

Glossaries created in a variety of databases, spreadsheets, and word-processors can be generated and localized in XML. Databases currently storing many megabytes of computer memory can be segmented in smaller, manageable XML chunks. Once in XML format, it makes it very portable and easy to handle. For example,

to add the different contexts of the word *Language* in a glossary, we can express it in XML as follows:

Example:

```
<Glossary context="programming">language</Glossary>
<Glossary context="english">language</Glossary>
<Glossary context="markup">language</Glossary>
<Glossary context="body">language</Glossary>
```

Meta-Data

Meta-data allow translators and applications to understand text even better by annotating pieces of text. These annotations are also called attributes. Localizers will have to work closely with each other and their customers to customize the DTD/schema and agree on the types of meta-data, names, and values. The quality of the source is decided not only by the actual translatable content, but also by structure and how meta-data are applied.

Below are a number of examples of how XML elements and meta-data can be applied for localization.

Example:

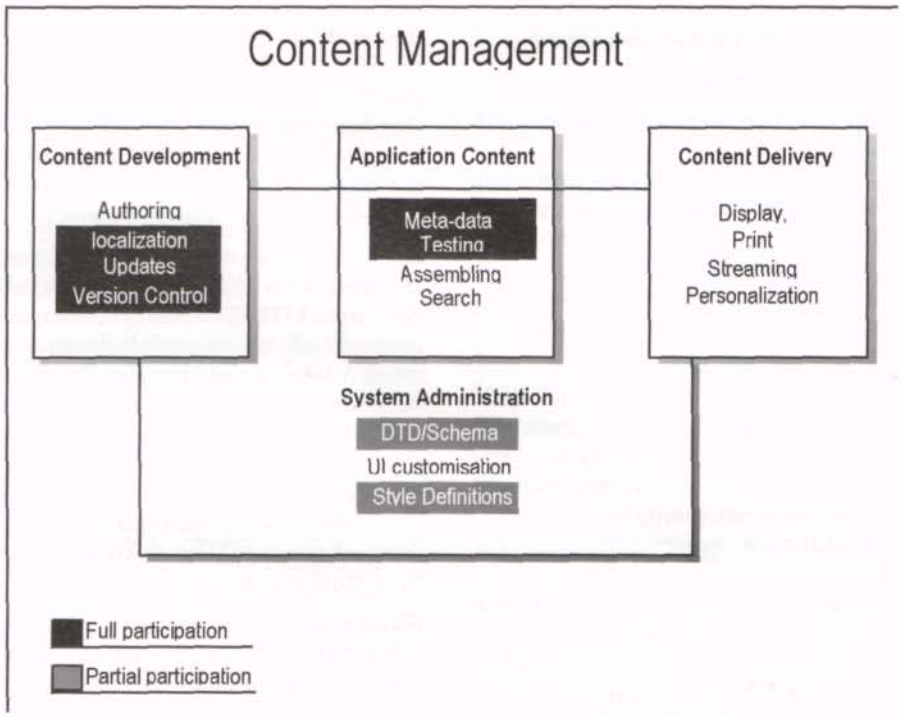
```
<btext status="in progress">This text is not
finalized.</btext>
<btext status="final">This text is final</btext>
<btext level="beginners">This text is targeted at
beginners only.</btext>
<btext level="advanced users">This text can be
understood by advanced users only.</btext>
<btext translation="No">Please do not translate this
text.</btext>
```

Glossaries created in a variety of databases, spreadsheets, and word-processors can be generated and localized in XML. Databases currently storing many megabytes of computer memory can be segmented in smaller, manageable XML chunks. Once in XML format, it makes it very portable and easy to handle.

```
<btext translation="FROnly">Please do not translate
this text into languages other than French.</btext>
```

This way we can query numerous types of information. Translating can commence based on the `status="final"` attribute while other areas are still in progress. Meta-data can be applied to a document, a section, a paragraph, a sentence, or a single word.

Meta-data also allow index entries to be categorized.



Example:

```
<Section level="beginners" mtopic="Metallic effects"
topic="How to do it" idx="stepbystep">

<Section level="advanced" mtopic="Water effects"
topic="How to do it" idx="stepbystep">
```

From this example you can generate and personalize an index that suits your level of expertise—a beginner's index, an advanced index, or both.

As you can imagine, the possibilities of automating translation processes are endless. The ability of recognizing and/or understanding text with meta-data will push translation technology to develop advanced recognition systems and make full use of it.

Content Management System (CMS) and XML

CMS and XML can fully complement each other. Information is treated as granular, manageable "chunks," or components, which can be accessed, updated, and assembled in real time. Thus, to build a document, the CMS might be queried on topics and reliability. This allows each topic to be created only once and it is therefore unique. Authors can tap existing "ready-to-go" topics to create documentation that comes with a new or updated product.

"Content management" has become a very broad term that is misused and overused. CMS addresses specific business issues and has many segments and subsegments. Localizers and their customers need to clearly define "content management" and choose the relevant segments. Content-management systems are usually a suite of authoring, workflow, assembling, and publishing tools that require a high level of system administration. These are costly and only a small segment may apply to localization vendors. For example, localizers will fully participate and contribute in the updates and meta-data implementation, while playing a minor role in style definition.

There are many XML editors on the market and all of them treat XML differently. They range dramatically in price, depending on your needs and budget—from simple color-coded flavors, to those with sophisticated editing and publishing features whereby markup is protected and validated on-the-fly. Most likely you'll see that XML will change the way you manage translation tools and memories.

XML Transformation

Publishing SGML documentation has always been a huge challenge. To format the SGML into displayable or printable media, a number of technologies such as FOSI and DSSSL are used. These are complex technologies that only a handful of engineers are willing to learn. This complexity has become a technical barrier for many companies to implement SGML. This is one of the many reasons XML was invented. XML documents do not include rendering information; in essence, it should be noted, an XML document must be transformed into another format. To transform XML files into documents with a layout for publishing, XSL or CSS style sheets are used. Much has been written about these standards. In essence, CSS is for formatting only, while XSL is for transformation and formatting. Naturally, you can only view XML in XML-enabled browsers like Internet Explorer 5 or Netscape 6. In a way, HTML is relatively easy to review linguistically as it is displayable, even if the styles are missing from it. Heavily coded XML makes it almost impossible to proof translated content.

Translating XML

XML files are text files and therefore easy to edit. Basic text editors like Notepad can be more than sufficient for translating smaller files. However, extra care is needed to preserve the markup. There are many XML editors on the market and all of them treat XML differently. They range dramatically in price, depending on your needs and budget—from simple color-coded flavors, to those with sophisticated editing and publishing features whereby markup is protected and validated on-the-fly. Most likely you'll see that XML will change the way you manage translation tools and memories:

- Translation memories may be categorized according to topics instead of books.
- Better leverage may be obtained, as the (text-only) XML markup is easier to manipulate.
- Localizable strings will be gathered and centralized from many other (non-XML) formats to a handful of XML files. This XML file is then localized and redistributed back to the source.

In Conclusion

Once documents are structured, the possibilities are endless. There are countless reasons why everyone is so excited about XML. However, to understand this excitement, we all need to do some homework on the concepts of XML.

Anyone handling XML, SGML, and/or content management projects will be faced with obscure terms and can easily become confused. Publishers and localization vendors will have to think differently about how documentation is created, and should waste no time in coming to grips with the basic concepts of structure and XML.

René Snel has worked for 10 years in the localization industry and has extensive multilingual publishing experience. He is now a member of the Technology Group in Lionbridge's Dublin office. Over the past four years, René has dealt with many complex SGML/XML localization and publishing challenges.