A MULTILINGUAL MARKUP TRANSLATION WEB-SERVICE*

An Entry Level Solution to Internationalize XML Markup Vocabularies

Alejandro Bia, Juan Malonda, Federico Botella
CIO, Universidad Miguel Hernández, Elche, Spain

Jaime Gómez
Departamento deLenguajes y Sistemas Informáticos, Universidad de Alicante, Alicante, Spain

Keywords: Internet services, XML markup, multilingual markup , internationalization.

Abstract: Markup is based on mnemonics (i.e. element names, attribute names and attribute values). These mnemonics have meaning, being this one of the most interesting features of markup. Human understanding of this meaning is lost when the encoder doesn't understand the language the mnemonics are based on. By "multilingual markup" we refer to the use of parallel sets of tags in various languages, and the ability to automatically switch from one to another. We started working with multilingual markup in 2001, within the Miguel de Cervantes Digital Library. By 2003, we have built a set of tools to automate the use of multilingual vocabularies (Bia et al, 2003). This set of tools translates both XML document instances, and XML document validators (we first implemented DTD translation, and then Schemas (Bia et al, 2004). First we translated the TEI tagset, and most recently the Dublin Core tagset (Bia et al, 2005) to Spanish, and Catalan. Other languages were added later[1]. Now we present a Multilingual Markup Website that provides this type of translation services for public use.

1 PREVIOUS WORK

At the time when we started this multilingual markup initiative in 2001 there were very few similar attempts to be found (Pei-Chi WU, 2000). Today they are still scarce (Bryan, 2002 and Cover, 2005).

Concerning document content, XML provides built-in support for multilingual documents: it provides the predefined *lang* attribute to identify the language used in any part of a document. However, in spite of allowing users to define their own tagsets, XML does not explicitly provide a mechanism for multilingual tagging.

* This work is part of the METASIGN project, and has been supported by the Ministry of Education and Science of Spain through the grant number: TIN2004-00779.

[1] Translations of the TEI tagset by: Alejandro Bia and and Manuel Sánchez (Spanish), Régis Déau (French), Francesca Mari (Catalan), Arno Mittelbach (German)

1.1 The Mapping Structure

We started by defining the set of possible translations of element names, attribute names, and attribute values to a few target languages (Spanish, Catalan and French). We stored this information in an XML translation mapping document called "tagmap", whose structure in DTD syntax is the following:

```
<!ELEMENT tagmap (element)+ >
<!ELEMENT element (attr)* >
  <!ATTLIST element
     en CDATA #REQUIRED
     es CDATA #REQUIRED
     fr CDATA #REQUIRED>
<!ELEMENT attr (value)* >
  <!ATTLIST attr
     en CDATA #REQUIRED
     es CDATA #REQUIRED
     fr CDATA #REQUIRED>
<!ELEMENT value EMPTY >
  <!ATTLIST value
```

Bia A., Malonda J., Botella F. and Gómez J. (2006).
A MULTILINGUAL MARKUP TRANSLATION WEB-SERVICE - An Entry Level Solution to Internationalize XML Markup Vocabularies.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 63-68
DOI: 10.5220/0001257900630068

```
        en CDATA #REQUIRED
        es CDATA #REQUIRED
        fr CDATA #REQUIRED >
```

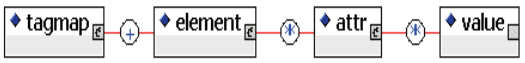

Figure 1: Structure of the original tagmap.xml file.

This structure is pretty simple, and proved useful to support the mnemonic equivalences in various languages. It was meant to solve ambiguity problems, like having two attributes of the same name in English, who should be translated to different names in a given target language. For this purpose, this structure obliges us to include all the attribute names for each element and their translations. The problem with this is global attributes, which in this approach needed to be repeated, once for each element. This made the maintenance of this file cumbersome. Sebastian Rahtz then proposed another structure (http://cvs.sourceforge.net/viewcvs.py/tei/I18N/teinames.xml), under the assumption that an attribute name has the same meaning in all cases, no mater the element it is associated to, and accordingly it would have only one target translation to a given language. This is usually the case, and although theoretically there could be cases of double meaning, as above mentioned, they do not seem to appear within the TEI. So the currently available "teinames.xml" file follows Sabastian's structure. Note that "element", "attribute" and "value" appear at the same level, instead of nested:

```
   <!ELEMENT i18n (element | attribute
| value)+>
   <!ELEMENT element (equiv | desc)* >
     <!ATTLIST element
        ident CDATA #REQUIRED >
   <!ELEMENT attribute (equiv | desc)*
>
     <!ATTLIST attribute
        ident CDATA #REQUIRED >
   <!ELEMENT value (equiv)* >
     <!ATTLIST value
        ident CDATA #REQUIRED >
   <!ELEMENT equiv EMPTY >
     <!ATTLIST equiv
        xml:lang CDATA #REQUIRED
        value CDATA #REQUIRED >
```

In 2004, we discussed the idea of adding brief text descriptions to each element, the same brief descriptions of the TEI documentation, but now translated to all supported languages. This would allow the structure to provide help or documentation services in several languages, as another multilingual aid. This capability was then added to the "teinames.xml" file structure, although the translations of the all the descriptions still need to be completed:

```
<!ELEMENT desc (#PCDATA) >
  <!ATTLIST desc
    xml:lang CDATA #REQUIRED >
```

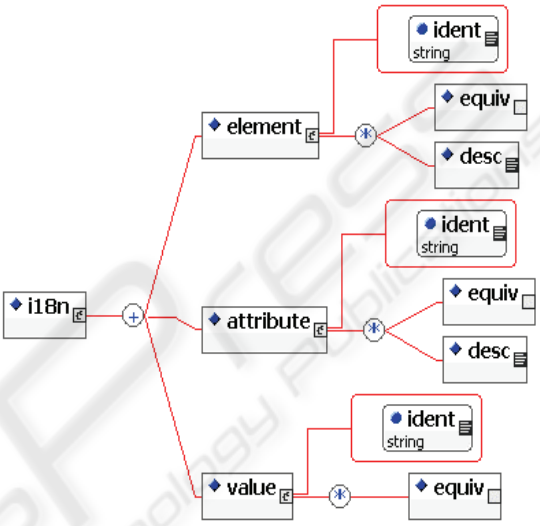

Figure 2: Structure of the teinames.xml file.

2 THE MULTILINGUAL MARKUP WEB SERVICE

By means of a simple input form, the markup of a structured file can be automatically translated to the chosen target language. The user can choose a file to process (see figure 3) by means of a "Browse" button.

Currently, only TEI XML document instances are allowed. In the near future, the translation of TEI DTDs, W3C-Schemas and Relax-NG Schemas will be added, and later, other markup and metadata vocabularies will be supported, like Docbook (Allen et al, 1997) and DublinCore (http://dublincore.org/).

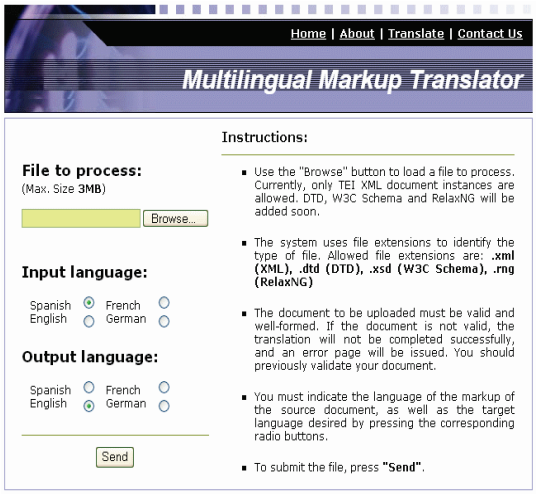

Figure 3: The Multilingual Markup Translator form.

The system uses file extensions to identify the type of file submitted. Allowed file extensions are: .xml for document instances, .dtd for DTDs, .xsd for W3C Schemas, and .rng for RelaxNG schemas.

The document to be uploaded must be valid and well-formed. If the document is not valid, the translation will not be completed successfully, and an error page will be issued. Once the source file has been chosen, the user must indicate the language of the markup of this source file, as well as the target language desired for the output. This is done by means of radio buttons.

It would not be necessary to indicate the language of the markup of the source file if it was implicit in the file itself. We thought of three ways to do this:

- To use the name of the root tag to indicate the language of the vocabulary of the XML document. In this way, TEI.2 would be standard English based TEI, TEIes.2 would indicate that the document has been marked up using the Spanish tagset, and in the same way TEIfr.2, TEIde.2, TEIit.2 would indicate French, German, and Italian, for instance.

- To add an attribute to the root element, to indicate the language of the tagset, for instance: <TEI.2 markupLang = "it"> would indicate that the markup is in Italian.

- Use the name of the DTD to indicate the language of the tagset. TeiXLite.dtd would be English, while TeiXLiteFr.dtd would be the French equivalent.

Option 3 is by far the worst method, since a document instance may lack a DOCTYPE declaration, and there may be lots of customized TEI DTDs everywhere with very different and unpredictable names. However, options 1 and 2 are reasonably good methods to identify the language of the markup. Consensus is needed to make one of them the common practice.

3 IMPLEMENTATION DETAILS

For the website pages we used JSP (dynamic pages) and HTML (static pages), and these are run under a Tomcat 5.5 web server. For the translations, we used XSLT, as described in (Bia et al, 2003)

3.1 Automatic Generation of Markup Translators Using XSLT

The XSLT model is thought to transform one input XML file into one output file (see figure 4), which could be XML, HTML, XHTML or plain text, and this includes program code. It does not allow the simultaneous processing of two input files.

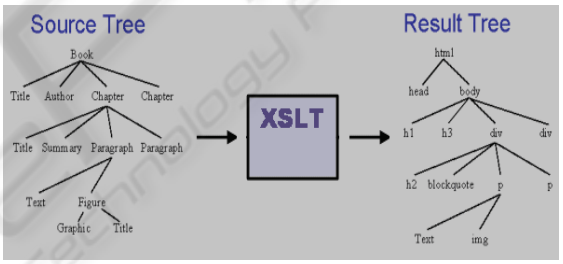

Figure 4: The XSLT processing model.

There are certain cases when we would like to process two input files altogether, like markup translation (see figure 5).

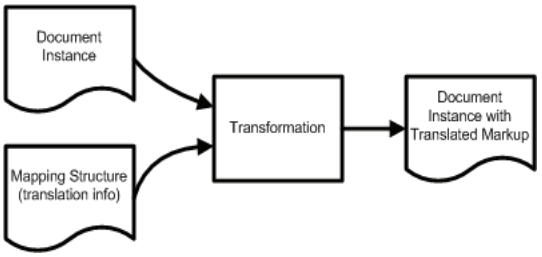

Figure 5: The ideal transformation required.

As XSLT does not allow this, two alternatives occurred to us, both comprising two transformation steps.

The first approach is to automatically generate translators. Douglas Schmidt said: "I prefer to write code that writes code, than to write code" (Schmidt, 2005). This is what we have done for the

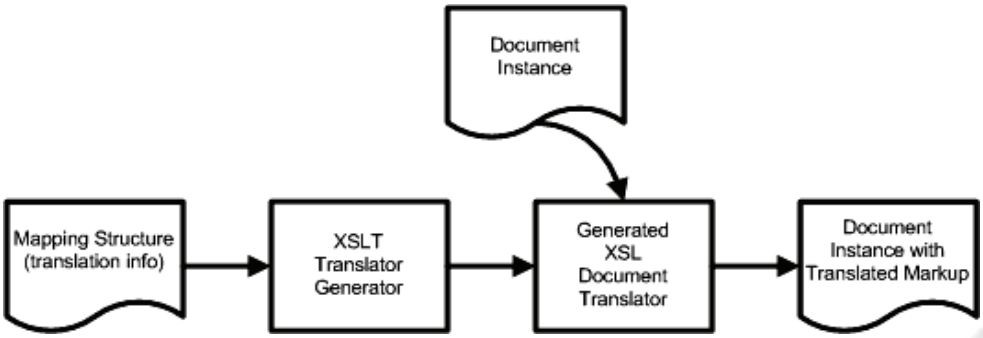

Figure 6: Pre-generation of a translating XSLT script, to then translate the document instance.

MMWebsite, i.e. to pre-process the translation map in order to generate an XSLT translation script which includes the translation knowledge embedded in its logic. Then this generated script can perform all the document-instance translations required. The mapping structure supports the language equivalences for various languages, so we should generate a translator for every possible pair of languages. Whenever the mapping structure is modified, a new set of translators must be generated. Fortunately, this is an automated process (se figure 6).

The other alternative would be to merge the two input files into a new single XML structure, and then to process such file which would contain both the XML document instance, and the translation mapping information (see figure 7). This implies joining the two XML tree structures as branches of a higher level root.

Although this approach may prove useful for some problems, we did not use it for the MMWebsite, because the file merging preprocessing must be done for each file to translate, increasing the web service response time. Using preprocessed translators instead proved to be a faster solution.

This limitation, which is proper of the XSLT processing model, could be avoided by using a standard programming language like Java instead.

3.2 How We Actually Do It

The mapping document which contains all the necessary structural information to develop the language converters is read by the transformations generator, which was built as an XSLT script. XSL can be used to process XML documents in order to produce other XML documents or a plain text document. As XSL stylesheets are XML, they can be generated as an XSL output. We used this feature to automatically generate both an English-to-local-language XSL transformation and a local-language to English XSL transformation for each of the languages contained in the multilingual translation mapping file. In this way we assured both ways convertibility for XML documents (see figure 8).

For each target language we also generate a DTD or a Schema translator. In our first attempts, this took the form of a C++ and Lex parser. Later, we changed the approach. Now we first convert the DTD to a W3C Schema, then we translate the Schema to the local language, and finally we can (optionally) generate an equivalent translated DTD. This approach has the advantage of not using complex parsers (only XSLT) and also solves the translation of Schemas. In our latest implementation, the user can freely choose amongst DTD, W3C Schema and RelaxNG, both for input and output, allowing for a format conversion during the translation process.

Many other markup translators can be built to other languages in the way described here.

4 CONCLUSIONS

Amongst the observed advantages of using markup in one's own language are: reduced learning times, reduction of errors and higher production. It may also help spread the use of XML vocabularies like DC, TEI, DocBook, and many others, into non-English speaking countries. Cooperative multilingual projects may benefit from the possibility of easily translating the markup to each encoder's language. Last, but not least, scholars of a given language feel more comfortable tagging their texts with mnemonics based on their own language.

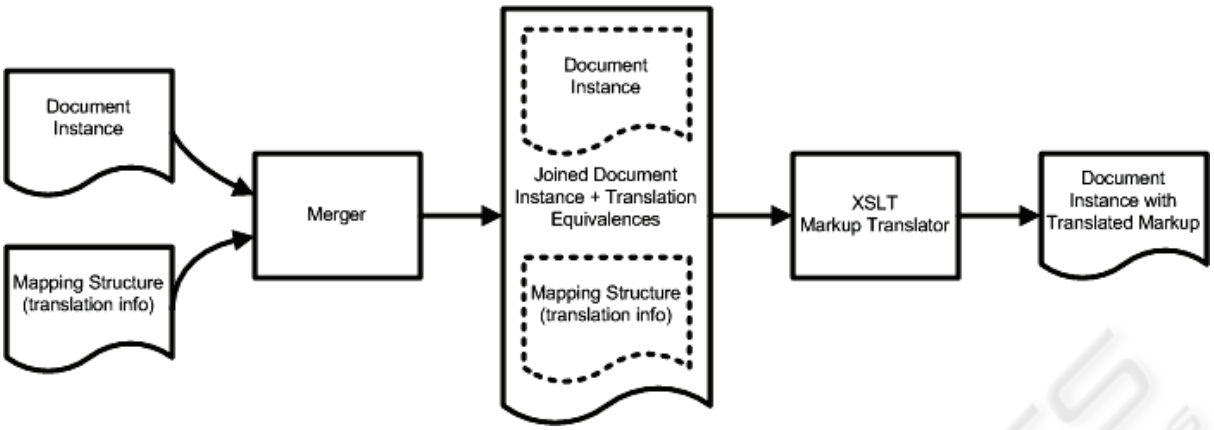

Figure 7: Merging the two files before applying XSLT.

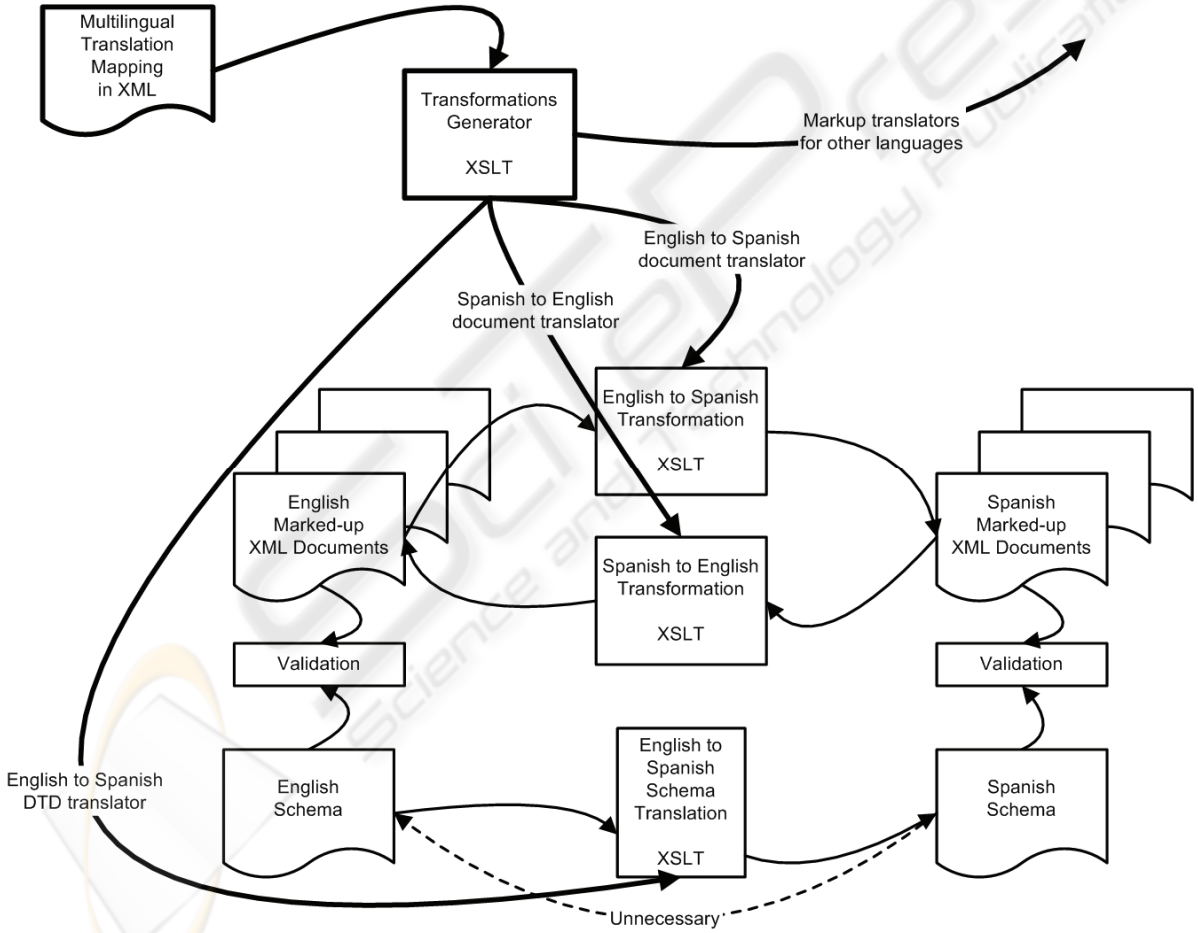

Figure 8: Schema translation using XSLT.

5 FUTURE WORK

Multilingual Help Services: As already said, brief descriptions for elements and attributes in different languages have been added to the mapping structure. This allows for multilingual help services, like generating a glossary in the chosen language of the elements and attributes used in a given document, or a given DTD/Schema. We are working on adding this feature.

REFERENCES

Allen, T., Maler, E. and Walsh, N., 1997. DocBook DTD, © 1992-1997 HaL Computer Systems, Inc., O'Reilly & Associates, Inc., Fujitsu Software Corporation, and ArborText, Inc, http://www.ora.com/davenport/

Bia, A., Sánchez-Quero, M. and Déau, R., 2003. *Multilingual Markup of Digital Library Texts Using XML*, TEI and XSLT. In XML Europe 2003 Conference and Exposition, Organized by IDEAlliance, 5-8 May 2003, Hilton Metropole Hotel, London, p. 53, http://www.xmleurope.com/

Bia, A., Sánchez-Quero, M., 2004. *The Future of Markup is Multilingual*, ACH/ALLC 2004: Computing and Multilingual, Multicultural Heritage. The 16th Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities, 11-16 June 2004, Göteborg University, Sweden, p 15-18, http://www.hum.gu.se/allcach2004/AP/html/prop119.html

Bia, A., Malonda, J. and Gómez, J, 2005. *Automating Multilingual Metadata Vocabularies*. In DC-2005: Vocabularies in Practice, Eva Mª Méndez Rodríguez (ed.), p. 221-229, 12-15 September 2005, Carlos III University, Madrid. ISBN 84-89315-44-2. http://dc2005.uc3m.es/

Bryan, J., 2002. *KR's Multilingual Markup*, TechNews Volume 8, Number 1: January/February 2002 http://www.naa.org/technews/TNArtPage.cfm?AID=3880

Cover, R., 2005. *Markup and Multilingualism*, last visited online 2005-4-25 at Cover Pages: http://xml.coverpages.org/multilingual.html

Pei-Chi WU, 2000. *Translation of Multilingual Markup in XML*, 2000 International Conference on the theories and practices of Electronic Commerce, Part II, Session 14, pages 21-36, Association of Taiwan Electronic Commerce, Taipei, Taiwan, October 2000. http://www.atec.org.tw/ec2000/PDF/14.2.pdf

Schmidt, D., 2005. Opening Keynote, MoDELS 2005: ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, Montego Bay, Jamaica, 2-7 October 2005.