Multilingual Markup Automation for Better Document Production and Retrieval

Bia Platas, Alejandro, Sánchez Quero, Manuel
Miguel de Cervantes Digital Library
University of Alicante
Apdo. correos 99,
Alicante, SPAIN, 03080
{alex.bia, manuel.sanchez}@cervantesvirtual.com

In this paper we will show the benefits of using multilingual markup schemes for large digitization projects like the Miguel de Cervantes Digital Library, and the advantages of using markup tags in one's own language, like the consequent increase in production, reduction of markup errors, and improved facilities for advanced XML based retrieval.

The solution presented here allows markup to be easily and automatically translated into many different languages, providing a useful way for building multilingual document repositories, with all the advantages this implies for document search and retrieval. We will also show the design of the multilingual markup structure that supports the mapping information to translate markup amongst many languages: currently English, Spanish and French, but the structure allows the addition of several other languages as well. In addition, we will explain the automatic generation of XSLT scripts to convert the markup of documents and DTDs/Schemas to various target languages. Finally, we will adduce the conclusions of the implementation of this project.

Recently, and as a consequence of this work, the TEI-Council (technical group that leads the development of the TEI markup scheme) agreed that it would be desirable for TEI to provide canonical translations for TEI element names, attribute names, and default values into as many languages as possible, whether by means of a distinct mapping document (the method proposed here) or by alternative methods. This Council charged an action to provide technical recommendations on definition and maintenance of a TEI term bank for document markup translation and for DTDs/Schemas multilingual-generation.

The solution presented here, allows for document markup to be easily and automatically translated to many different languages, providing a useful way of developing multilingual document repositories, with all the advantages this has for information retrieval purposes.

Keywords: Automatic Multilingual Markup Translation, Digital Libraries, Information Retrieval.

INTRODUCTION

The Miguel de Cervantes Digital Library[1] is the biggest electronic publishing project in Spain, and perhaps the biggest digital library of Spanish texts on the Internet, currently with more than 10000 entries in its catalogue. It produces an average of 150 XML (Extensible Markup Language) digital texts per month, most of which are Spanish classics from the 12th century up to these days, comprising a wide variety of subjects and styles such as poetry, narrative, drama, history, geography, law, etc. [1] These texts are used both by the casual reader and by specialized researchers that take advantage of the power of complex structural markup.

In a project like this, the amount and quality of production depends highly on technology. Even slight changes in critical aspects of the production technology involved produce important changes in times, costs and the quality of the final output.

MARKUP, MEANING AND MULTILINGUALISM.

In 1998 Robin Cover wrote: *"How does XML help with the encoding of information at the semantic level? Or does it? New users sometimes refer to XML as semantic markup, and may be heard to praise XML for its ability to express semantic clarity through markup. ... Someone who uses a text editor to examine an XML document -- comparing it to an ancient WordStar file, to a*

comma-delimited text file, to Postscript, or to any document using a procedural or presentational markup language -- will readily judge the XML document more meaningful with respect to the information objects represented by text. The markup itself is a form of 'metadata', explaining to us what the constituent elements are (by name), and how these information objects are structured into larger coherent units." [2]

In spite of currently preferring predicate logic as more suitable than conventional DTDs for semantic purposes [3], Sperberg-McQueen et.al. [5] supported the usefulness of markup as a source of meaning. They rhetorically asked and answered themselves: *"The function of markup is not random. Markup has meaning. What does it mean to have meaning? How does markup have meaning? Why worry about this question?" "For better markup language documentation, for better QA (verification), for better automated processes (translation, normalization, query), to provide a way to survey current practice (relevance for software developers) ... and because it's interesting"*. *"How does markup mean? Because markup means something, ... we know certain things. I.e. because we see certain markup, we are allowed (licensed) to make certain inferences"*. and concluded that: the meaning of markup is the set of inferences it licenses. [5]

So one of the key aspects of structural markup is the meaning it conveys, which depends on our ability to understand it. Understanding XML tags is key to correctly delimit complex text structures for further automated processing. This understanding may be compromised when tag names (elements, attributes and attribute values) are in a foreign language. For instance, the following example shows something as simple as `<title type="main">`, but in Russian:

```
<название главный="тип">
```

Hard to read, isn't it? Unless you know Russian, of course.

Our digital library is a multidisciplinary project where specialists from different study fields (philologists, computer scientists, librarians, sociologists, etc.) work together in cooperation. The largest group of specialists in the library is the proof-reading and markup team (about 40 persons), comprised of specialists from different humanities fields, none of them related to the English language. It is in this area where the necessity and importance of translating the original English markup into one's own language (Spanish in our case) is made evident. For encoders unfamiliar with English, tag names are meaningless[2], *"just like a set of complicated separators used in the document"*, as Wu says [6].

We learned from practice that using a tagset in a foreign language, compared to using a tagset in our own language, increases the learning time and reduces the quality and amount of digital text production, since tag names are mnemonics that may sound familiar to English speakers but are hard to understand and memorize by users of other languages. Giving our encoders the possibility of applying tags in Spanish has increased the amount and quality of digital text production.

PHASES OF THE PROJECT

1. We took the decision to use a standard markup vocabulary like TEI[3]
2. We initially used it in its original language.
3. We began translating it into Spanish.
4. Creation of prototype translation tools for conversion of TEI documents from English to Spanish.
5. Implementation of automatic conversion programs to translate both XML marked-up documents and also DTDs/Schemas.
6. Design of generators of these markup translators to generalize the solution to other languages as well.
7. Now we are in the process of building other TEI tagsets and translations for other languages (a TEI term bank).

Convinced as we are of the value and advantages of using standards we have chosen the TEI (Text Encoding Initiative) tagset which is a de facto standard at least within the English-literature scholar community. After using it successfully for sometime, we embarked in the project of translating TEI element names, attribute names and attribute values into Spanish. Finally we developed the translation tools to grant automatic conversion to and from the main TEI English core. These automatic conversion programs translate not only the markup of XML documents but also the corresponding DTDs. Then we designed automatic generators that could easily generate all the necessary markup translators for XML documents, DTDs and Schemas needed to translate to and from several languages. This generators produce all the necessary XSLT translators from the mapping information contained in the tag mapping XML file (see figure 1). We only needed to build two generators, one to generate the XML document translators, and the other to generate the DTDs or Schemas translators. At runtime the generators are

passed command-line parameter values indicating the source and the target languages of the translator to be generated. In this way, and with only two relatively small XSLT scripts we are able to generate in batch mode the whole set of document and DTDs/Schemas translators to and from many different languages.

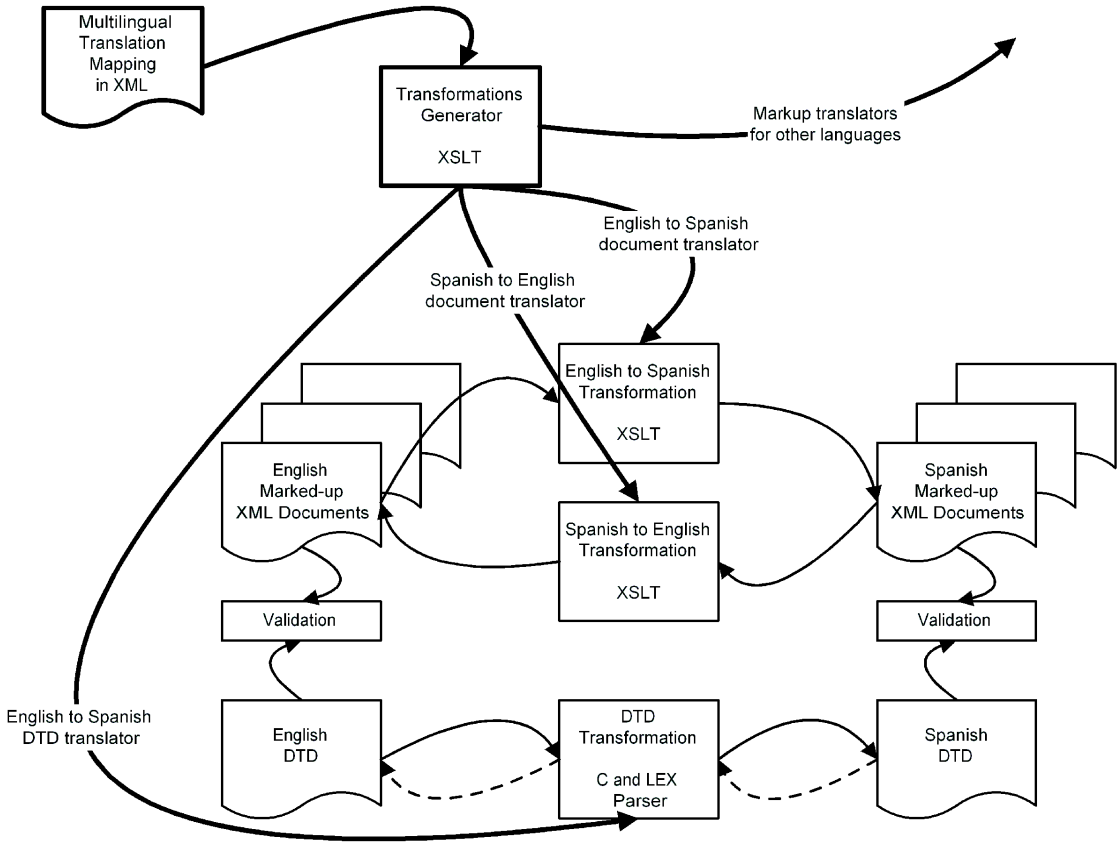

Fig. 1: Automatic generation of markup translators: This figure describes the generation of XSL transformations and C++ parsers to convert English markup and DTDs to Spanish.

Now the only remaining task is to add new sets of terms to the tag-map to cover other new languages. The ultimate purpose is to have many official translations of the TEI tagset, but one core version (the original English one). The automation of the language translation of the tags is vital to assure easy interchangeability of documents amongst projects using different languages. In this way, and from the structural and semantic point of view, the tagset is the same, only the names change.

We also believe that having multilingual versions of a given tagset, like TEI, can facilitate their introduction in many parts of the world like Latin America. This may be of special interest for digital libraries and digital publishers worldwide. The main reason for this initiative is that markup schemes usually are defined in English and there is a large community of users who do not use the English language so fluently and then lose its meaning. If the markup scheme is translated into the users' language, the process of assimilating and controlling its use will be accelerated and the production of marked-up texts will increase, with the corresponding reduction of costs.

So one of the purposes of this initiative is the introduction of standard markup schemes (generally in English) into non-English-speaking communities where XML for electronic publishing is still uncommon. We know of many projects in Spanish speaking countries that prefer to create markup vocabularies of their own instead of having to use an existent tagset in English, no matter how good, proven and well supported it is. This has many disadvantages, compared to using a standard vocabulary: difficulties concerning interchangeability, lack of other user's support, and not being able to use already existent tools and stylesheets.

Using a standard markup vocabulary but in one's own language offers the best of both worlds: enhanced understanding of the semantics of the markup scheme, plus interchangeability amongst users of the same standard, plus user's community support, plus collaborative development of tools and guides of good practice, to mention a few.

AUTOMATIC GENERATION OF MARKUP TRANSLATORS

We started by defining the set of possible translations of element names, attribute names, and attribute values into the different target languages. We stored this information in an XML multilingual translation mapping document. An example of this document and its DTD follow.

TRANSLATION MAPPING DOCUMENT FOR ENGLISH, SPANISH AND FRENCH (SMALL SAMPLE):

```
<TAGMAP>
...
 <ELEMENT eng="body"
          esp="cuerpo"
          fra="corps">
 </ELEMENT>
...
 <ELEMENT eng="div0"
          esp="div0"
          fra="div0">
  <ATTR eng="lang"
        esp="lengua"
        fra="langue">
  </ATTR>
  <ATTR eng="type"
        esp="tipo"
        fra="type">
   <VALUE eng="news"
          esp="noticias"
          fra="nouvelles"/>
   <VALUE eng="suggestions"
          esp="sugerencias"
          fra="sugestions"/>
   <VALUE eng="biblnews"
          esp="novedades"
          fra="publications"/>
  </ATTR>
 </ELEMENT
...
 <ELEMENT eng="p"
          esp="parrafo"
          fra="paragraphe">
  <ATTR eng="align"
        esp="alinear"
        fra="aligne">
   <VALUE eng="left"
          esp="izq"
          fra="gauche"/>
   <VALUE eng="right"
          esp="der"
          fra="droite"/>
   <VALUE eng="center"
          esp="centro"
          fra="centre"/>
   <VALUE eng="justify"
          esp="justificar"
          fra="justifie"/>
  </ATTR>
  <ATTR eng="indent"
        esp="sangria"
        fra="retraitpositif">
   <VALUE eng="left"
          esp="izq"
          fra="gauche"/>
   <VALUE eng="right"
          esp="der"
          fra="droite"/>
   <VALUE eng="both"
          esp="ambas"
          fra="lesDeux"/>
   <VALUE eng="none"
          esp="ninguna"
          fra="aucune"/>
  </ATTR>
  <ATTR eng="specialindent"
        esp="sangriaespecial"
        fra="retraitnegatif">
   <VALUE eng="none"
          esp="ninguna"
          fra="aucune"/>
   <VALUE eng="firstline"
          esp="primeralinea"
          fra="premiereLigne"/>
   <VALUE eng="french"
          esp="francesa"
          fra="francaise"/>
  </ATTR>
 </ELEMENT>
...
</TAGMAP>
```

DTD FOR THE ABOVE FILE:

```
<!ELEMENT TAGMAP (ELEMENT)+ >
<!ELEMENT ELEMENT (ATTR)* >
<!ATTLIST ELEMENT
     eng CDATA #REQUIRED
     esp CDATA #REQUIRED
     fra CDATA #REQUIRED>
<!ELEMENT ATTR (VALUE)* >
<!ATTLIST ATTR
     eng CDATA #REQUIRED
     esp CDATA #REQUIRED
     fra CDATA #REQUIRED>
<!ELEMENT VALUE EMPTY >
<!ATTLIST VALUE
     eng CDATA #REQUIRED
     esp CDATA #REQUIRED
     fra CDATA #REQUIRED>
```

This 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 [4]. XSL can be used to process XML documents in order to produce other XML documents or a plain text document. As XSL scripts are XML, they can be generated as an XSL output. In this way, and for each of the languages contained in the multilingual translation mapping file, we produced both an English to local-language XSL transformation and a local-language to English XSL transformation. In this way we assured both ways convertibility for XML documents.

We also generated (in our first versions) a DTD translator from English into each target language in the form of a parser written in C++ and Lex. Take

into account that DTDs are not XML compliant and hence cannot be transformed using XSLTs. So it is for this we used the XSL capability of producing plain text, but in the form of a C++ program in this case. Later we changed this approach as explained below.

We only produce a one way translation from the English DTD into a target-language DTD, since we assumed that the DTD would be first built in the original XML vocabulary language (English) and then translated into the local language, and not the other way around. We saw no need to translate the local-language DTD back to English (that's why we use a dashed line in the diagrams), but this is a transformation that could very easily be generated in the same way if the need arises, allowing for maintenance and modifications of the DTD to be done in the local language and then translated into English. Just as the English DTD can be used to validate the English-marked-up set of XML documents, the local language DTD can be used to validate the local-language marked-up set of files.

This translator generation process is shown for Spanish as the target language in figure 1. Many other markup translators can be built to other languages in the same way. In our tests we played with English, Spanish and French, being able to generate transformations to translate to and from any pair of these languages, although our idea is to translate to and from the original tag-set's language (English), which should be used as the standard file transfer language amongst projects.

At a later stage, it occurred to us that converting DTDs to Schemas will help us avoid the need of building Lex/C++ parsers, since Schemas are also XML and they can be easily parsed using XSLT (see figure 2). In this way no custom-built C parsers are needed, leading to a simpler and more elegant solution within the world of the XML family of tools (XML, XSL, XPATH...).

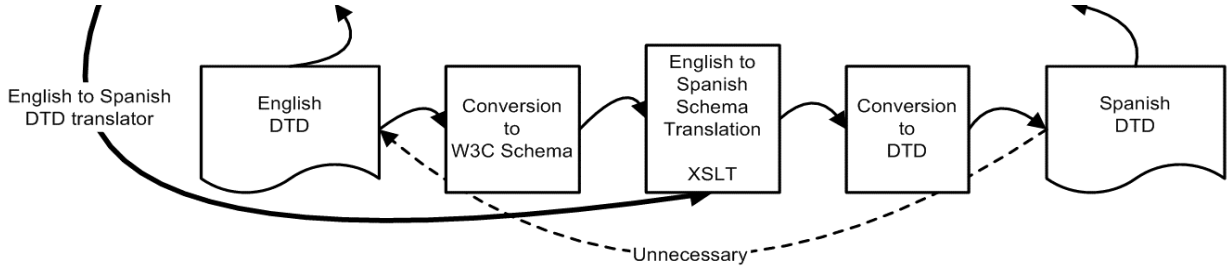

Fig. 2: Converting DTDs to Schemas to avoid the need of a C parser: As Schemas are also XML, they can be parsed with XSLT. No custom built C parser is needed.

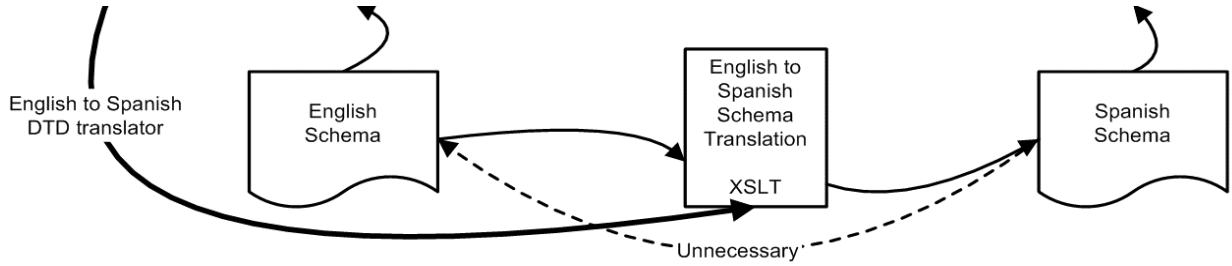

Fig. 3: W3C Schemas can also be translated: The approach of converting DTDs to Schemas has also provided a solution to the problem of translating Schemas.

By using this approach we also solved the conversion of Schemas to other languages with no additional effort (see figure 3), as this was implicit in the previous solution (figure 2). This will be useful for encoders who prefer the advantages of the newer Schemas to the old DTDs as a newer and richer way to define and control markup.

In favor of DTDs we can say that they are accepted by all XML/SGML editors and are part of the XML standard. They are also more compact than Schemas, but here the advantages end. Schemas, on the other hand, are larger in size (RelaxNG offers also a DTD-like compact mode to overcome this), but clearer to read, easier to process and validate (they are XML), and add new useful features like data types. Our implementation of the multilingual markup translation tools currently supports W3C Schemas, but Relax NG will surely be added soon.

SOME ISSUES ON USAGE

With a minimalist approach, we think local-language markup should be used almost only for creation and maintenance purposes (i.e. to tag, edit,

and correct the XML documents). There are cases where it should also be used, as for XML searches when the search-engine interface allows the user to enter queries using markup elements, attributes and attribute values, which would be better understood in the local language. If the user has to use tag names in a foreign language the semantic advantage is lost. A translating interface can also be used for multilingual XML search engines.

For automated processing and document interchange we think it is more convenient to use markup in the language of the original standard. In this way, previously existent stylesheets that have been designed for the original language of the markup vocabulary need not be translated into the local tagset language of the document for processing, but the document markup can be translated into the original tagset instead before applying the stylesheet. These saves the time and effort of having to adapt the existent tools (like stylesheets) to the new language of the markup. In this way, all the existent stylesheets (our own and those available from other users), can be still applied to our documents no matter what language we might have chosen for the markup of our texts (they can transparently and quickly be turned back to English at any time). However, encoding, maintenance or searches can be done using tagging in our native language for enhanced understanding.

This approach can be argued, and there may be users who prefer to translate everything (including stylesheets) into the local language. Although this is not strictly necessary, it can be easily done.

A risk to be avoided, if the advantages of using a standard widely accepted markup scheme are to be preserved, is the development of alternative markup schemes in different languages which may evolve independently from the central standard.

As it became clear from the start of our digital library project, "a principle of good practice in information technology is to avoid duplication of data, and this includes texts" [1], so another risk to avoid is the existence of duplicates of a document, a well known problem with serious consequences in data handling. We believe that only one source document should exist, no matter the language we chose for its markup, and all the rest of the copies, if any, should be considered disposable duplicates or temporaries to be discarded. In this way, maintenance and modifications should only be performed on the unique source document. This a versioning control problem that escapes the scope of this project, but it is a problem that should deserve high attention from developers of multilingual marked-up documents and tools.

IMPACT ON DOCUMENT SEARCH AND RETRIEVAL

The use of XML for digital publishing allows for complex searches based on semantic tags. Searches based on the structure of the document, which are not possible with ordinary relational database architectures are becoming more and more common.

Advanced XML searches benefit from the fact of using tags in one's own language, in our case Spanish. In this way, not only the contents of the documents is in Spanish (our case), but also the tagging (element names, attribute names and attribute values). This allows for complex XML searches to be performed in a more understandable way, closer to natural language.

For example, we can search for an historic document from Spain, under the name of "Palafox y Mendoza" and that mentions Mexico in the title. If the query is performed using the original TEI markup it will look like this:

```
"ESPAÑA" in <country> and
"MEXICO" in <title> and
"PALAFOX", "Y" and "MENDOZA"
in <name>
```

This is an unnatural mixture of English and Spanish. Much more natural to the user would be a query on Spanish texts using Spanish markup, like this:

```
"ESPAÑA" en <pais> y
"MEXICO" en <titulo> y
"PALAFOX", "Y" y "MENDOZA"
en <nombre>
```

TEI ACCEPTANCE

This work on multilingual markup was presented during a TEI Council meeting in Oxford (May 16th, 2003). In discussion, the Council agreed that it would be desirable for TEI to provide canonical translations for TEI element names, attribute names, and default values into as many languages as possible, whether by means of a distinct mapping document (as proposed here) or by additional information in the ODDs[4]. Such mappings would be best developed by user communities rather than centrally by the TEI, which might lead to consistency and conformance problems. It was agreed to form a taskforce to look into ways of defining and maintaining a TEI term bank.

If this project succeeds, TEI will be the first widely used markup vocabulary to become multilingual.

PREVIOUS WORK

In a couple of words: not much.

Concerning document contents, XML does have 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.

It is not easy to find, in the available literature, antecedents of attempts to use multilingual tagging, let alone of building tools to automate the translation process. We assume there must have been various isolated attempts to translate or build customized markup vocabularies using different local languages to solve specific problems[5], but very little of this has been published. We don't know of standard (or widely-used) markup vocabularies ever been translated to other languages, let alone to have been made multilingual and the translation process been fully automated. However, we found an interesting article from Pei-Chi Wu [6], that addresses the problem of translating a tagset to another language[6] for easier understanding and more accurate markup. As this author states: *"In Extensible Markup Language (XML), users can even define their own markups using local languages. These are widely accepted practices to make documents more easily grasped by local users"*. This paper addresses the issue of multilingual markup, proposes a bilingual translation process, and discusses its potential applications to electronic commerce.

They describe a prototype built with Java and MSXML (Microsoft XML) for the translation process, which is based on parallel equivalents DTDs (one for each language version of the markup vocabulary) instead of the predefined XML mapping file for translation we use. In their process they first build the mapping file by comparison of the source and target DTD, and then parse the documents changing tag names. Comparing to our approach, they do not support schemas, their method proposes to build the mapping table every time a file is processed, although their prototype does not do so (their mapping table is built by hand), they do not translate attribute names nor defaulted attribute values, and they do not generate translators for XML documents and DTDs or Schemas. However, their work is an interesting antecedent to read, where they highlight the usefulness of this type of markup translation tools for electronic commerce.

CONCLUSIONS

By applying the tools and methods described here to produce documents with TEI-XML tagging in Spanish within the Miguel de Cervantes DL, we realized that:

- Learning times were noticeably reduced.
- Production times were also reduced, along with an increase in markup quality (less errors while structuring documents). Encoders showed themselves satisfied and more confident in their task.
- When using markup in one's own language, the meaning of markup is not lost.
- Advanced XML searches benefit from the fact of using tags in one's own language, in our case Spanish.

We also believe that:

- Cooperative multilingual projects may benefit from the possibility of easily translating the markup into each encoder's language.
- Sometimes new non-standard vocabularies are developed just because it seams comparatively easier than learning a standard vocabulary in a foreign language. Having the possibility of using a standard vocabulary in one's own language plays against developing a new custom vocabulary to fulfill a local markup requirement. This may help spread the use of XML vocabularies like TEI or DocBook in non-English speaking countries.
- Spreading the use of standard markup vocabularies is good for document interchangeability.

We started by making a translation of TEI into Spanish. We ended up with a general set of tools to convert any markup vocabulary (not only TEI) to many languages. There may be better implementations to solve this problem. However, the problem is an interesting one and deserves to be solved. The development and use of multilingual tagsets should be spread to become common practice.

FUTURE WORK

This translation tools could also be implemented as a client-server Web service. The TEI Council decided to charge an action to implement TEI multilingual markup translations by means of a Web service, and also to build a multilingual term bank for translation mappings.

DocBook and other standard vocabularies with high semantic values for structural markup may follow this initiative.

BIBLIOGRAPHY

[1] BIA, Alejandro, and PEDREÑO, Andrés (2001): "The Miguel de Cervantes Digital Library: the Hispanic voice on the Web", in *LLC (Literary and Linguistic Computing) journal*, Oxford University Press, 2001, vol. 16, n. 2, pages 161-177, ISSN: 0268-1145.

[2] COVER, Robin (1998): XML and Semantic Transparency (in Cover Pages). October 23, 1998. Revised November 24, 1998. http://www.oasis-open.org/cover/xmlAndSemantics.html

[3] DUBIN, David, SPERBERG-MCQUEEN, Michael, RENEAR, Allen, and HUITFELDT, Claus (2002): "A logic programming environment for document semantics and inference", *ACH/ALLC 2002: New Directions in Humanities Computing. The 14th Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities*, pages 34-36, University of Tuebingen, Germany , 24-28 July, 2002.

[4] KAY, Michael (2000): *XSLT Programmer's Reference*, Wrox Press, 2000, 1102 Warwick Road, Acocks Green, Birmingham, B27 6BH, UK, 1st. ed., ISBN 1-861003-12-9,

[5] SPERBERG-MCQUEEN, C. M., HUITFELDT, Claus, and RENEAR, Allen (2000): "Meaning and Interpretation of Markup not as simple as you think", in *Extreme Markup Languages*, Montreal, 15 August 2000.

[6] WU, Pei-Chi (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.[7]

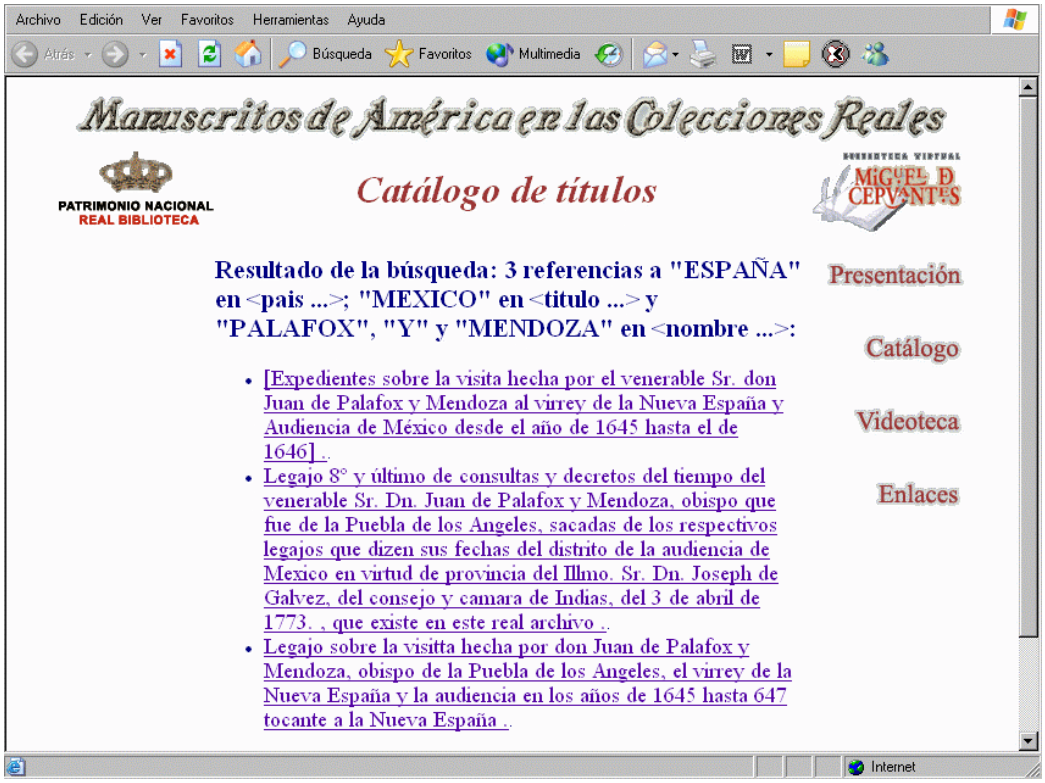

Fig. 4: Example of an XML search performed on Spanish texts using Spanish TEI tags.

[1] http://cervantesvirtual.com/

[2] Wu [6] supports the same idea but for Chinese encoders, for whom Chinese tags are as he says "very clear".

[3] http://www.tei-c.org/

[4] The TEI literate programming system (jocularly named ODD, for 'One Document Does it all') as originally specified is documented in an internal TEI working paper (www.tei-c.org/Vault/ED/edw29.sgm).

[5] Wu [6] gives some examples using Chinese.

[6] In this case the translations were from English to Chinese.

[7] URL: http://www.atec.org.tw/ec2000/PDF/14.2.PDF