Automating Multilingual Metadata Vocabularies

Alejandro Bia
Universidad Miguel Hernández
Tel: +34 96 6658542
Fax: +34 966658715
abia@umh.es

Juan Malonda
Universidad Miguel Hernández
Tel: +34 96 6658542
jmalonda@umh.es

Jaime Gómez
Univesidad de Alicante
Tel: +34 96 590 3306
jgomez@ua.es

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 have a good command of the language the mnemonics are based on. By "multilingual markup" we refer to the use of tags built with mnemonics in one's own language, but still following the rules of the original markup vocabulary. In this paper we show the benefits of using multilingual markup vocabularies, especially in large digital library projects, and we describe our work to automate the use of multilingual vocabularies, including the translation of DC to Catalan, French, German and Spanish. ?•

Keywords:

Multilingual markup, Dublin Core, TEI [1] , XML.

1. Benefits of Multilingual Vocabularies

We started this idea of multilingual markup at the Miguel de Cervantes Digital Library. The largest group of workers there is by far the proof-reading and markup team, comprised of about 40 persons. They are graduates from different humanities fields, none of them

?• 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.

The TEI (Text Encoding Initiative) is a very complete as well as powerful markup vocabulary, both for text and metadata, originally based on SGML but now available in XML format. Different subsets of this markup vocabulary expressed in DTD, Relax NG or W3C Schema formats, can be obtained from a service called "Roma" at he TEI website: http://www.tei-c.org/

related to the English language. It is in this area where the necessity and importance of translating the original English markup into the local language (Spanish) is made evident. 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 encoders the possibility of applying tags in Spanish has increased the amount and quality of digital text production. After successfully using XML-TEI for sometime, we embarked in the project of translating TEI element names, attribute names and attribute values to Spanish. Then 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 validators (DTDs, XML Schema, RelaxNG). Then we repeated the experience with Catalan, German and French. Now we are in the process of building other TEI tagsets and translations for several other languages.

• This work is part of the METASIGN project, andhas been supported by the Ministry of Education andScience of Spain through the grant number:
TIN2004-00779.

[1] The TEI (Text Encoding Initiative) is a verycomplete as well as powerful markup vocabulary,
oth for text and metadata, originally based onSGML but now available in XML format. Differentsubsets of this markup vocabulary expressed in DTD,Relax NG or W3C Schema formats, can be obtainedfrom a service called "Roma" at he TEI website:http://www.tei-c.org/

https://doi.org/10.23106/dcmi.952108244

The purpose is to have many official translations of the TEI tagset, but one core version (the original one). More recently we performed the translation of Dublin Core elements with their descriptions to Catalan, French, German and Spanish. Tag translation automation 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 remains the same, only the names change. We also believe that having multilingual versions of given tagsets, like DC or TEI, can facilitate their acceptance and use in many parts of the world like Latin America where the use of XML for electronic publishing is still uncommon. This may be interesting for digital libraries and digital publishers worldwide, but especially within the European Union where multilingual projects can benefit in a remarkable way.

2. Introduction

Markup allows us to define the structure of both text and metadata in a way that can be processed by computer programs and also understood by humans. Human understanding is hampered when the tags are based on a foreign language. This is usually the case for non-English speakers. For instance, a Spanish encoder that doesn't know English will find difficult and error prone to apply or understand DC markup using the original DC mnemonics based on the English language. This applies to all the widely used metadata and hypertext markup vocabularies, which are based on English (e.g. DC, MODS, METS, RDF, teiHeader for metadata, and TEI, Docbook and even the popular HTML for hypertext). In our own experience, an equivalent version of any markup vocabulary can be developed based on Spanish, Catalan, French and almost any other language. The tools for translating back and forth to the original "canonical form" in English can be built automatically and can be applied in a transparent and easy way, as we will further explain. When we build an equivalent markup vocabulary in a language other than English, the structural properties and constraints of the original markup scheme remain the same in the target language. Only the terms used for elements, attributes and attribute values in both schemas [2] and document instances are different, making the document structure remarkably clearer for the non-English encoder. In the following sections, we will describe our implementation of multilingual markup based on the automatic generation of translating scripts using XSLT [3] for both document instances and schemas as well. We will also discuss other alternative implementations to the

one proposed that look promising. Finally, we will present the conclusions of the implementation and use of this technology within the Miguel de Cervantes Digital Library. We will also comment on the creation of a TEI Multilingual Markup Special Interest Group (TEI-MM-SIG) and the involvement of the TEI META Workgroup in the development and full implementation of a multilingual term-bank for the TEI.

3. Markup, meaning and multilingualism.

One of the key aspects of structural markup is the meaning it conveys, which depends on our ability to understand it. In 1998 Robin Cover wrote: *How does XML help with the encoding of information at the semantic level? ... 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 ... 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*. [1] Sperberg-McQueen et.al. [2] supported the usefulness of markup as a source of meaning:

The function of markup is not random. Markup has 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. 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*. So understanding XML tags is essential 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.

4. Previous Work

At the time when we started this multilingual markup initiative in 2001 there were few similar attempts to be found [4]. Today they are still scarce [5, 6]. Concerning document contents, XML does have built-in support for multilingual documents: it

[2] By "schemas" in lowercase, we refer to all typesof XML document validators, including DTDs, XML Schemas, RelaxNG, and others.

[3] Extensible Stylesheet Language Transformation

https://doi.org/10.23106/dcmi.952108244

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 , 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 [4], that addresses the problem of translating a tagset to another language for easier understanding and more accurate markup. As this author states: "In Extensible Markup Language (XML), users can even define their own markup 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.

5. Automatic generation of markup translators

We started by defining the set of possible translations of element names, attribute names, and attribute values to different target languages. We stored this information in an XML multilingual translation mapping document. An example of this document is shown in Table 1. 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 [3]. 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. 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 (see figure 1). 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 (see figure 2). This approach has the advantage of not using complex parsers (only XSLT) and also solves the translation of Schemas, which is an interesting goal in itself (see figure 3). 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, as demonstrated by our tests with Catalan and French.

6. Usage and implementation alternatives

We think that markup in the local-language should only be used for tasks which require human intervention, like creation and maintenance of documents. 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, processing tools like stylesheets need not be translated to the local language, but the document translated to the original tagset instead. An alternative, and perhaps the most effective implementation of multilingual markup, could be a translating interface integrated into an XML editor. In this way, we would have virtual views of the document with markup in different languages that could be toggled at the touch of a button, but without actually having to translate the document file. An implementation like this is possible today, but can only be done by the software companies who build XML editors. This built-in solution would not require the DTD/Schema to be translated. An editor like this would need to load the mapping information (tag-map), as well as the DTD and the document instance (see figure 4). A compromise solution that can be integrated into some XML editors by expert users is to build macros that automatically

https://doi.org/10.23106/dcmi.952108244

apply the translation to local language on opening the document, and the translation back to English on closing. This would not be as handy as a one-key language-toggling solution, but can be implemented by users. Additional macro programming would also be required for translation before validation and before applying further processing like XSLT. If multilingual markup becomes a common practice, the mapping structure with the name equivalences for markup translation could well be included as part of a new form of Schema. In any case, this use should be specified and formally integrated into the XML family of standards.

7. Conclusions

Amongst the observed advantages of using markup in one's own language are: reduced learning times, reduction of errors and higher production.

-Are the advantages of using general and widespread vocabularies like DC and TEI lost? Not at all. The two main advantages of using a general metadata/markup vocabulary are **document interchangeability** and **community support** (which includes training and tool sharing). Since markup terms can be very easily and automatically translated to the original English tagset, interchangeability is not lost and rendering tools like XSLT scripts could still be used unchanged after markup translation back to English. Training materials, however, may need to be translated or adapted, but this is not due to the use of multilingual markup but to the need of non-English-speaking encoders to have documentation in their language.

-In our experience, learning times were noticeably reduced. -Production times were also reduced, along with an increase in markup quality. Encoders showed themselves satisfied and more confident in their task.

-By using markup in one's own language, the meaning of markup is not lost, and the document structure suddenly becomes clearer. -Librarians, scholars and students showed approval for being able to handle documents with markup in the same language of the text.

-Cooperative multilingual projects may benefit from the possibility of easily translating the markup to 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 fulfil a local markup requirement. This may help spread the use of XML vocabularies like DC, TEI, DocBook, and many others, in non-English speaking countries.

-Spreading the use of standard markup vocabularies is good for metadata and document interchangeability.

8. Future work

Interpretation of Markup not as simple as A special interest group on multilingual markup (TEI-MM-SIG) has been created within the TEI Consortium to exploit and expand the benefits of using multilingual markup. During its first meeting at the 2003 TEI annual meting, the idea, tools and possibilities of multilingual markup have been introduced, and the objectives of the group have been established.The full implementation of multilingual support for other vocabularies should be carried out. Here we propose a case study based on DC (see Tables 2 and 3 below). The technical possibilities, limitations and challenges of multilingual markup should be further studied. There are many aspects to be discussed and decisions yet to be made. To give an example, there may be problems to overcome if we want to build mnemonics using accented or oriental characters.

9. Bibliography

[1] Robin Cover, Cover Pages XML and Semantic Transparency. October 23, 1998. Revised November 24, 1998. http://www.oasis-open.org/cover/xmlAnd Semantics.html

[2] C. M. Sperberg-McQueen, Claus Huitfeldt and Allen Renear, Meaning and Interpretation of Markup not as simple as you think, in Extreme Markup Languages, Montreal, 15 August 2000.

[3] Michael Kay, XSLT Programmer's Reference, Wrox Press, 2000, 1102 Warwick Road, Acocks Green, Birmingham, B27 6BH, UK, 1st. ed., ISBN 1-861003-12-9,

[4] 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)

[5] John Bryan, KR's Multilingual Markup, TechNews Volume 8, Number 1: January/February 2002

https://doi.org/10.23106/dcmi.952108244

(http://www.naa.org/technews/TNArtPage.cfm?AID=3880)

[6] Robin Cover, Markup and Multilingualism, last visited online 2005-4-25 at Cover Pages: http://xml.coverpages.org/multilingual.html

Table 1: DC translation mapping for English, Spanish, Catalan, French and German (abridged):

```
<?xml version='1.0' encoding='iso8859-1' ?> <!—
* Dublin Core Basic Elements Translation Map.
* Basic Dublin Core Elements translated to Spanish, Catalán, French and German
* Descriptions of elements translated to Spanish an Catalán.
- Translation to Spanish (Alejandro Bia and Juan A. Malonda Campos).
- Translation to Catalán (Juan A. Malonda Campos).
- Translation to French (Alejandro Bia).

         - Translation to German (Raphael Schnuc and Alejandro Bia). —>

<dcNames>
         <element ident = "title">
         <equiv lang = "es" value = "título"/>
         <equiv lang = "ca" value = "títol"/>
         <equiv lang = "fr" value = "titre"/>
         <equiv lang = "de" value = "Titel"/>


           <desc lang = "en">A name given to the resource.</desc>
           <desc lang = "es">Nombre dado al recurso.</desc>

       </element>

         <element ident = "creator">
         <equiv lang = "es" value = "creador"/>
         <equiv lang = "ca" value = "creador"/>
         <equiv lang = "fr" value = "créateur"/>
         <equiv lang = "de" value = "Ersteller"/>


            <desc lang = "en">An entity primarily
   responsible for making the content of the
   resource.</desc>        <desc lang = "es">Entidad
   principal responsable de hacer el contenido del
   recurso.</desc>    </element>

... [SOME ELEMENTS ARE OMMITED HERE]

         <element ident = "rights">
         <equiv lang = "es" value = "derechos"/>
         <equiv lang = "ca" value = "drets"/>
         <equiv lang = "fr" value = "droits"/>
         <equiv lang = "de" value = "Urheberrecht"/>


  <desc lang = "en">Information about rights held in and over the resource.</desc>
  <desc lang = "es">Información de derechos sobre el recurso.</desc> </element>

</dcNames>
```

https://doi.org/10.23106/dcmi.952108244

Table 2: A Simple Dublin Core DTD using tags in Spanish:

```
    <!–#DOCUMENTATION: DTD based on Simple DC XML Schema, 2002-10-09      by Pete Johnston
    (p.johnston@ukoln.ac.uk),      Carl Lagoze (lagoze@cs.cornell.edu), Andy Powell
    (a.powell@ukoln.ac.uk),      Herbert Van de Sompel (hvdsomp@yahoo.com).

 Translated to Spanish (2005-4-10) by Alejandro Bia
(abia@umh.es) and Juan Malonda (jmalonda@umh.es).    –>

    <!ELEMENT registroDC ((título | creador | tema | descripción | editorial | colaborador |
    fecha | tipo | formato | identificador | fuente | idioma | relación | cobertura |
    derechos)*)>

<!ELEMENT título (#PCDATA)> <!ELEMENT
creador (#PCDATA)> <!ELEMENT tema
(#PCDATA)> <!ELEMENT descripción
(#PCDATA)> <!ELEMENT editorial
(#PCDATA)> <!ELEMENT colaborador
(#PCDATA)> <!ELEMENT fecha (#PCDATA)>
<!ELEMENT tipo (#PCDATA)> <!ELEMENT
formato (#PCDATA)> <!ELEMENT
identificador (#PCDATA)> <!ELEMENT
fuente (#PCDATA)> <!ELEMENT idioma
(#PCDATA)> <!ELEMENT relación
(#PCDATA)> <!ELEMENT cobertura
(#PCDATA)> <!ELEMENT derechos
(#PCDATA)>
```

Table 3: Example of an XML metadata record using Dublin Core in Spanish:

```
<?xml version=”1.0” encoding=”ISO-8859-1”?> <!DOCTYPE
registroDC SYSTEM “dublincore_es.dtd”[
    <!ELEMENT registroDC ((dc:título | dc:creador | dc:tema | dc:descripción | dc:editorial
    | dc:colaborador | dc:fecha | dc:tipo | dc:formato | dc:identificador | dc:fuente |
    dc:idioma | dc:relación | dc:cobertura | dc:derechos)*)> ]>

    <registroDC>  <!– Ejemplo de Dublin Core en castellano–> <dc:título
    xml:idioma=”es”>Marcado Multilingüe en Bibliotecas Digitales</dc:título>
    <dc:creador>Alejandro Bia</dc:creador> <dc:creador>Juan Malonda</dc:creador>
    <dc:relación>See also Multilingual Markup...</dc:relación> <dc:tema
    xml:idioma=”en”>Multilingual markup vocabularies</dc:tema> <dc:fecha
    xsi:tipo=”dct:W3CDTF”>2005-04-10</dc:fecha>
</registroDC>
```

https://doi.org/10.23106/dcmi.952108244

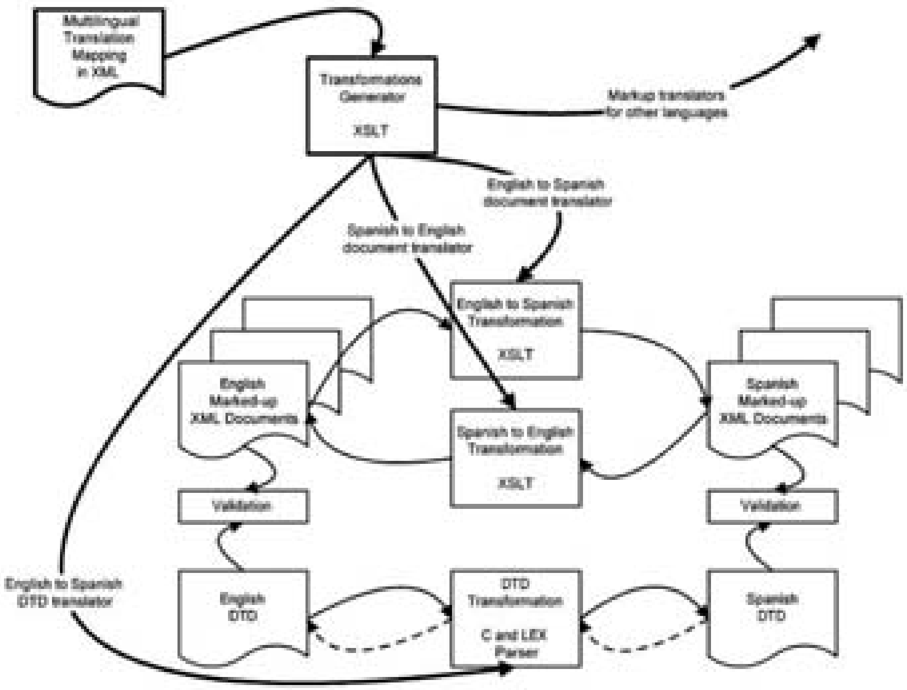
Multilingual Translation Mapping in XML
Transformations Generator XSLT
Markup translators for other languages
English to Spanish document translator
Spanish to English document translator
English to Spanish Transformation XSLT
Spanish to English Transformation XSLT
English Marked-up XML Documents
Spanish Marked-up XML Documents
Validation
Validation
English to Spanish DTD translator
English DTD
DTD Transformation C and LEX Parser
Spanish DTD

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.

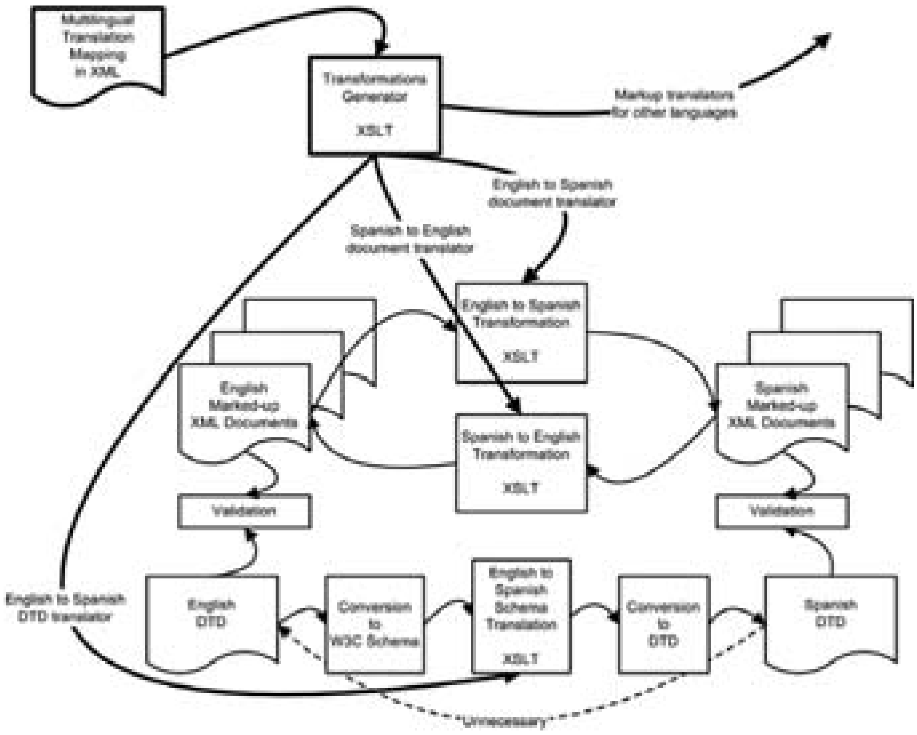
Multilingual Translation Mapping in XML
Transformations Generator XSLT
Markup translators for other languages
English to Spanish document translator
Spanish to English document translator
English to Spanish Transformation XSLT
Spanish to English Transformation XSLT
English Marked-up XML Documents
Spanish Marked-up XML Documents
Validation
Validation
English to Spanish DTD translator
English DTD
Conversion to W3C Schema
English to Spanish Schema Translation XSLT
Conversion to DTD
Spanish DTD
Unnecessary

Fig. 2: DTD translation using XSLT and an intermediate Schema.
This figure describes the same process of figure 1 but using only XSLT.

https://doi.org/10.23106/dcmi.952108244

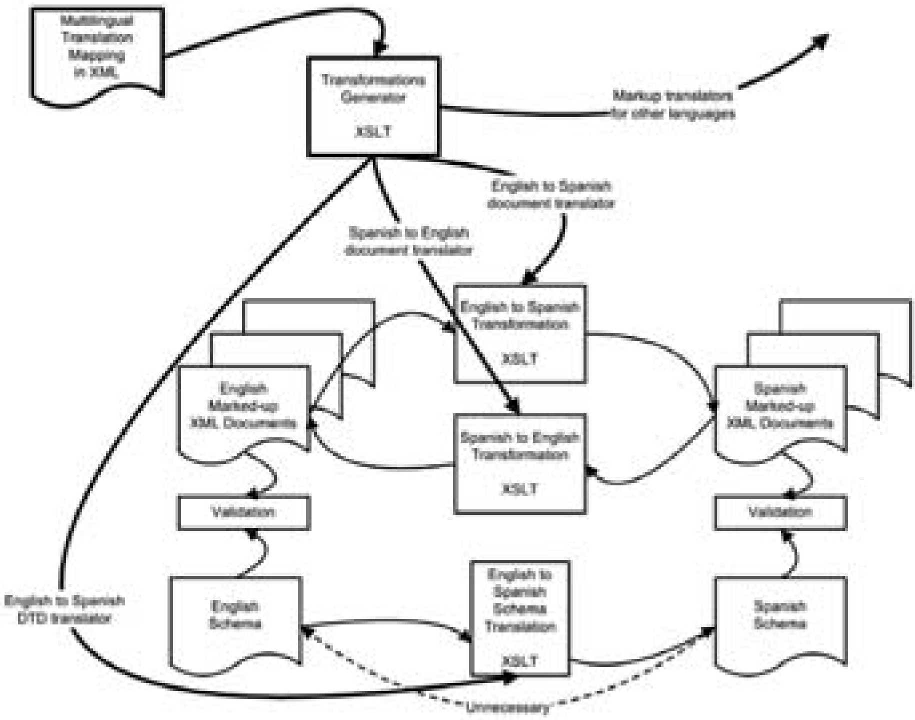
Multilingual Translation Mapping in XML
Transformations Generator
XSLT
Markup translators for other languages
English to Spanish document translator
Spanish to English document translator
English to Spanish Transformation
XSLT
Spanish to English Transformation
XSLT
English Marked-up XML Documents
Spanish Marked-up XML Documents
Validation
Validation
English to Spanish DTD translator
English Schema
English to Spanish Schema Translation
XSLT
Spanish Schema
Unnecessary

Fig. 3: Schema translation using XSLT.
The solution shown in figure 2 for translating DTDs by first converting them to Schemas and then using XSLT, implicitly solves the translation of Schemas.

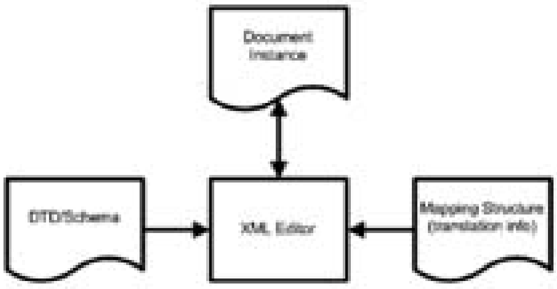

Fig. 4: XML Editor for Multilingual Markup.
Apart from the document instance and the DTD/Schema for validation, a mapping structure with information for the translation is also required.

https://doi.org/10.23106/dcmi.952108244