How to Import an Existing XML Dictionary Into the Papillon Plateform

Mathieu Mangeot-Lerebours

National Institute of Informatics (NII)
Hitostubashi 2-1-2-1913, Chiyoda-ku, Tokyo 101-8430, Japan
mangeot@nii.ac.jp

Abstract. The Papillon project aims at building a multilingual lexical database for extracting dictionaries. To avoid starting from scratch, we plan to recuperate existing dictionaries(marking explicitely all the possible information) and integrate them into the Papillon one (converting the structure into the Papillon structure). The plateform used to manage the Papillon dictionary [2] can also manage any other dictionary as soon as it is encoded in XML. This paper presents the steps necessary to recuperate an existing dictionary and to install it on the Papillon plateform . It will be then available for consultation through the Papillon consultation interface.

1 Introduction

The Papillon project aims at building a multilingual lexical database for extracting dictionaries. To avoid starting from scratch, we plan to recuperate existing dictionaries(marking explicitely all the possible information) and integrate them into the Papillon one (converting the structure into the Papillon structure). The plateform used to manage the Papillon dictionary [2] can also manage any other dictionary as soon as it is encoded in XML. This paper presents the steps necessary to recuperate an existing dictionary and to install it on the Papillon plateform . It will be then available for consultation through the Papillon consultation interface.

2 Conversion of the structure into XML

2.1 Conversion of the XML entities

The conersion of XML entities must be done before any other conversion. The & must be converted before the others otherwise the & will create an infinite loop.

```
"&" => "&amp;"
"'" => "&apos;"
"<" => "&lt;"
">" => "&gt;"
```

2.2 Information Tagging

In this step, we need to encode explicitely a maximum of the dictionary information. If the original structure of the dictionary is flat, we have to create a new deep structure for this dictionary with new XML tags. we do not need to use the Papillon XML tags

neither to convert into the Papillon structure. This conversion is another step. Nevertheless, we encourage people to use as much as possible the CDM/DML tags used as common pointers into heterogeneous structures. For more informaiton about CDM/DML, please read the LREC paper [1].

If the dictionary has been generated automatically by a program (it means that there are no errors in the structure) and/or the structure is very simple you can convert the dictionary directly using a regexp tool like perl scripts or a word processor like Word or BBEdit.

If the dictionary is hand written and/or the structure is complicated, you may use the RECUPDIC recuperation method with the Hgrammar compilator developped by Hai Doan during his Ph.D. Thesis (please read the paper from Aree Teeraparbseree [3] for more information on this tool).

3 Conversion of the encoding

You don t need to convert your file if the encoding is already accepted by the XML SAX parser (eg: UTF-8, ISO-8859-1).

For other encodings, you must first find a tool to convert them to utf-8. For MacOs, the tool cyclone is free for any use (please see http://homepage.mac.com/tkukiel/cyclone.html for more info). For UNIX/LINUX plateforms, you can use the uniconv program or the perl module cjkvconv.pl. The next lines show how to convert a document from the Japanese EUC encoding into the UTF-8 encoding.

```
>uniconv -o output-UTF-8 -I EUC_JP -O UTF8 input-EUC
>cjkvconv -ie -ou8 < input-EUC > output-UTF-8
```

If you convert an XML file, be careful to change the encoding of the XML header to UTF-8:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

4 Writing the metadata files

4.1 Editors and validation

The integration of a dictionary into the Papillon plateform is done via XML documents describing the metadata of the dictionary. These documents are described by the DML XML schema. This schema and XML examples are available online at http://www-clips.imag.fr/geta/services/dml/dml.xsd.

To write an XML metadata file, you can use any text editor. Nevertheless, we recommend to use an XML editor like XML Spy to avoid XML syntax errors. XML Spy for PC available at http://www.xmlspy.com for a 30 days free evaluation version.

The best tool is an XML schema validating editor. The editor will check online the validity of your documents thanks to the URL indicated at the root element of the XML documents with the `xsi:schemaLocation` XML attribute :

```
<dictionary-metadata
    xsi:schemaLocation="http://www-clips.imag.fr/geta/services/dml"
```

If your editor does not check the validity online, you can use the online W3C XML Schema validator: http://www.w3.org/XML/Schema.

4.2 The dictionary metadata file

The dictionary metadata file describes the macrostructure of the dictionaries, ie the organisation and links between the volumes composing the dictionary. It describes also the metainformation available on the dictionary: creation date, authors, etc.

All the reference addresses that point to other local files of the dictionary must be relative. ie the path must start from where the dictionary-metadata file is located. eg: indicate <volume-metadata-ref href= myDict_fra-metadata.xml /> instead of <volume-metadata-ref href= /Users/Projects/Dicts/myDict/myDict_fra-metadata.xml />.

The languages are represented by the ISO 639-2/T (T for terminology) standard which uses 3 letters code instead of two letters code to indicate the name of the languages eg eng for English, fra for French, jpn for Japanese, deu for German, etc. We extended the standard with a few codes for special puroses. For a complete list of language codes, please refer directly to the DML schema.

The file must be valid with the <dictionary-metadata> element of the dml XML schema. An example of the French-English-Malay monodirectional dictionary metadata is available online at http://www-clips.imag.fr/geta/services/dml/FeM-metadata.xml. It should be used as a starting point for writing new dictionary metadata files.

Below is a commented example of a dictionary metadata file for a bilingual bidirectional French-Vienamese dictionary.

```
<dictionary-metadata
```

First, you indicate the namespaces of the DML schema for automatic validation purpose. You should not modify these lines.

```
    xmlns='http://www-clips.imag.fr/geta/services/dml'
    xmlns:d='http://www-clips.imag.fr/geta/services/dml'
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www-clips.imag.fr/geta/services/dml
    http://www-clips.imag.fr/geta/services/dml/dml.xsd"
```

Then, you describe the meta-information on the dictionary with several attributes. The category of the dictionary can be monolingual , bilingual or multilingual .

```
category="bilingual"
```

Creation date of the dictionary.

```
creation-date="21/01/02 00:00:00"
```

Installation date of the dictionary. Date where the metadata file is created.

```
installation-date="21/01/02 00:00:00"
```

Full name of the dictionary.

```
fullname="Duc's Vietnamese-French Dictionary"
```

Short nickname of the dictionary.

```
name="VietDict"
```

Owner of the dictionary.

```
owner="Vietnam, under GPL"
```

Macrostructure type of the dictionary. Can be monodirectional , bidirectional , pivot or mixed .

```
type="bidirectional">
```

The element languages lists the languages present in the dictionary.

```
<languages>
 <source-language d:lang="fra"/>
 <source-language d:lang="vie"/>
 <target-language d:lang="fra"/>
 <target-language d:lang="vie"/>
</languages>
```

The element contents describes with a text the contents of a dictionary

```
<contents>general vocabulary but a bit old French</contents>
```

The element domain describes the domain of a dictionary e.g. : general, medecine, computer science, etc. Maybe it could be a closed list...

```
<domain>general</domain>
```

The element source describes from where does the dictionary come from, who gave it to the database.

```
<source>Ho Ngoc Duc - Uni-Leipzig, www.informatik.uni-leipzig.de/
~duc </source>
```

```
 <authors>group of vietnamese students and researchers,
www.saigon.com/~vietdict/project.html, converted by Jerome GODARD
jerome@nii.ac.jp </authors>
```

The element legal describes the legal rights attached to the use of this dictionary e.g. :research purpose only, public, GPL, open source, etc.

```
 <legal>under GPL</legal>
```

The element comments is used for general text comments on a dictionary.

```
 <comments> we have a letter from Duc confirming Papillon can use
those data </comments>
```

The element administrators list all the administrators of the dictionary, ie people allowed to modify the files.

```
 <administrators>
  <user-ref name="Mathieu.Mangeot"/>
 </administrators>
```

The element volumes lists all the volumes/files of a dictionary with an xlink URL hypertex reference.

```
 <volumes>
  <volume-metadata-ref
  xlink:href="VietDict_fra_vie-metadata.xml"
  source-language="fra"/>
  <volume-metadata-ref
  name="VietDict_vie_fra" xlink:href="VietDict_vie_fra-metadata.xml"
  source-language="vie"/>
 </volumes>
```

The element links indicates the links between the volumes files in the dictionary. It is typically used to describe a pivot dictionary. This dictionary does not have specific links between its volumes, therefore the element link is empty.

```
  <links/>
```

We indicate here the element links for the Papillon pivot dictionary.

```
  <links>
   <link xlink:to="Papillon_eng" xlink:from="Papillon_axi"
oriented="true"/>
   <link xlink:to="Papillon_fra" xlink:from="Papillon_axi"
oriented="true"/>
   <link xlink:to="Papillon_jpn" xlink:from="Papillon_axi"
oriented="true"/>
   <link xlink:to="Papillon_lao" xlink:from="Papillon_axi"
oriented="true"/>
   <link xlink:to="Papillon_tha" xlink:from="Papillon_axi"
oriented="true"/>
```

```
   <link xlink:to="Papillon_vie" xlink:from="Papillon_axi"
oriented="true"/>
  </links>
```

The element other-files gives info on other files about the dictionary.

```
  <other-files>
   <file xlink:href="vietdict.html">Installation Instructions of the
original Duc Software</file>
  </other-files>
```

The element xsl-stylesheet references an xsl stylesheet file with an xlink. This file is used to display the entries in HTML. The section 5 describes how to write an XSL stylesheet for a dictionary.

```
  <xsl-stylesheet xlink:href="VietDict-view.xsl"/>
</dictionary-metadata>
```

4.3 The volume metadata file(s)

The volume metadata file is used to describe parts of the structure of the entries with the CDM elements [1] and metadata information specific to the volume.

The file must be valid with the <volume-metadata> element of the dml XML schema. An example of the French-English-Malay volume metadata is available online at http://www-clips.imag.fr/geta/services/dml/FeM_eng_fra_msa-metadata.xml. It should be used as a starting point for writing new volume metadata files.

Below is a commented example of the French->Vietnamese volume metadata file of a bilingual bidirectional French-Vienamese dictionary.

```
<volume-metadata
```

First, you indicate the namespaces, same as above.

```
   xmlns='http://www-clips.imag.fr/geta/services/dml'
   xmlns:d='http://www-clips.imag.fr/geta/services/dml'
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www-clips.imag.fr/geta/services/dml
   http://www-clips.imag.fr/geta/services/dml/dml.xsd"
```

Then, you describe the meta-information on the volume with several attributes.

The attribute location indicates if the volume is stored locally ie you will upload it to the Papillon server database (value local) or if it is a remote server, ie the papillon application will connect to the remote server with a wrapper (value remote). In our case, we will use only the local value.

```
   location='local'
```

Creation date of the volume.

```
creation-date="21/01/02 00:00:00"
```

Number of entries in the volume.

```
hw-number="47106"
```

Installation date of the dictionary. Date where the metadta file is written.

```
installation-date="21/01/02 00:00:00"
```

Name of the volume.

```
name="VietDict_fra_vie"
```

This is the name of the database table that will be created for this volume. Some databases (like Postgres) does not accept characters like _ or - so the best is to use only the 26 alphabet small letters.

```
dbname="vietdictfravie"
```

Version number of the volume.

```
version="1"
```

Source and target languages of the volume.

```
source-language="fra"
target-languages="vie">
```

The Common Dictionary Markup CDM elements are used as common pointers into an unknown structure. These pointers will be used for parsing and database table keys for indexing and querying. The elements cdm-volume, cdm-entry and cdm-headword are compulsory. For each CDM element, you indicate the dictionary equivalent information. The attribute element indicates which is the correspondent element in the dictionary structure that tags the information. If this information is tagged also by an attribute, the attribute attribute indicates the name of the attribute. Here, the entry id is indicated with the attribute id of the element entry . For more information about the CDM elements, please read [1].

```
<cdm-elements>
 <cdm-volume element="dictionary"/>
 <cdm-entry element="article"/>
 <cdm-entry-id element="article" attribute="id"/>
 <cdm-headword element="word"/>
 <cdm-pos element="part-of-speech"/>
 <cdm-translation d:lang="vie" element="trans"/>
```

The element corpus is a special CDM element. It indicates if all the text of the entry can be searched as a corpus or not.

```
 <cdm-corpus d:delay="10s"/>
```

```
 </cdm-elements>
```

The element administrators list all the administrators of the volume, ie people allowed to modify the files.

```
 <administrators>
  <user-ref name="Mathieu.Mangeot"/>
 </administrators>
```

Finally, the element volume-ref references with an xlink the volume file containing the list of entries.

```
 <volume-ref xlink:href="vietDict_fra_vie-UTF8.xml" source-
language="fra"/>
</volume-metadata>
```

5 Writing the XSL stylesheets

Your dictionary is now encoded in XML and ready to be uploaded. In order to be displayed into the Papillon server, you need to write an XSL stylesheet that will be applied by default on the dictionary entries.

You can write a general XSL stylesheet for all the volumes of your dictionary and then an XSL stylesheet for each of the volumes for elements depending on the volumes.

The stylesheets will be applied in cascade by default if the user did not select one precise stylesheet. The dictionary stylesheet wil be applied first, then the volume stylesheet and finally the default Papillon stylesheet.

for the moment, you have to write the XSL stylesheets by hands but we plan in the future to automatically generate the default XSL stylesheet from the CDM description in the metadata file.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<xsl:stylesheet version="1.0"
  xmlns:d='http://www-clips.imag.fr/geta/services/dml'
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/1999/xhtml">
```

The output is also XML because other stylesheets will be applied before the HTML display.

```
<xsl:output method="xml" encoding="utf-8" indent="no"/>
```

These two template are the default template. If no special template is applied, it copies the element and attributes as is to the output. You shouldnot modify them.

```
<xsl:template match="*" priority="-1">
```

```
<xsl:element name="{name()}"><xsl:apply-templates
select="@*|*|text()"/></xsl:element>
</xsl:template>

<xsl:template match="@*">
<xsl:copy/>
</xsl:template>
```

The root template sould also not be modifed.

```
<xsl:template match="/">
<div>
  <xsl:apply-templates/>
</div>
</xsl:template>
```

Then, for each element of the dictionary, you specify a template. For the CDM-elements you indicated above, your template should add the corresponding CDM elements (here prefixed by d:).

```
<xsl:template match="fem-entry">
<d:entry>
 <xsl:apply-templates/>
</d:entry>
</xsl:template>
```

You can also include any text string or HTML tag as soon as it is XML compliant (ie an open tag has always a closing tag).

```
<xsl:template match="FRENCH_CAT">
<xsl:text> </xsl:text>
<br />
<d:pos>
 <xsl:apply-templates/>
</d:pos>
</xsl:template>

</xsl:stylesheet>
```

6 Upload into te Papillon Server

Once all your files are ready and validated, you olny need to upload the dictionary metadata file. The other files (Volume metadata files, data entries, XSL stylesheets) are linked from the dicitonary metadata file and hence will be automatically uploaded.

For the moment, the application is only able to upload files accessibles via a public URL, so all your files must be accessible via a public URL, ie be on a public web server.

To upload the dictionary metadata file and all the other files, you have to be in the admin group. You logon to the papillon server and access to the AdminDictionaries.po page. There, if you want to upload all the files ie also volumes and entries, you check the appropriate checkboxes. Then you push the OK button and the upload starts.

If you want to manage separately volume metadata files, you can access to the Admin-Volumes.po page. You still need to be in the admin group. The same is applicatble for the entries files manageable from the AdminEntries.po page.

7 Conclusion

The papillon plateform is built to handle any dictionary as soon as it is encoded in XML. One could use its own proper Papillon application to manage its own proper dictionaries. The plateform is still at the early stages of the developments.

We plan in the future to implement a compilator for the SUBLIM extended language that could generate the whole code of tha application and the description of the dictionaries.

8 References

1. Mathieu Mangeot-Lerebours (2002) An XML Markup Language Framework for Lexical Databases Environments: the Dictionary Markup Language. LREC Workshop on International Standards of Terminology and Language Resources Management, Las Palmas, Islas Canarias, Spain, 28 May 2002, pp 37-44.

2. Mathieu Mangeot-Lerebours (2001) Environnements centralis s et distribu s pour lexicographes et lexicologues en contexte multilingue. Th se de nouveau doctorat, Sp cialit Informatique, Universit Joseph Fourier Grenoble I, jeudi 27 septembre 2001, 280 p.

3. Aree Teeraparbseree A Practical Guide to Lexical Data Acquisition with Recupdic Proceedings of Papillon 2002, Tokyo, Japan, 16-18 July 2002,To be published, 17 p.