Proposal Changes for the Papillon Monolingual XML Schema

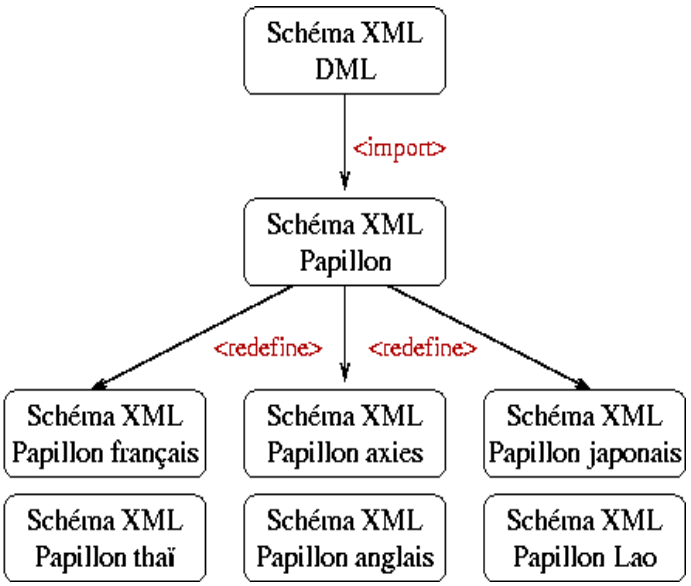
Mathieu Mangeot-Lerebours

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. This paper describes the Papillon monolingual lexie structure with an example and propose some changes for spotted problems.

1 Introduction

The Papillon XML entries structure are described with XML schemata.



The basic schema is the DML schema. This schema describe general elements and attributes used to encode lexical databases in XML. It allows one to encode users, groups, APIs, dictionaries, modifications histories and entries. The DML is an XML representation and extension of the SUBLIM languages from Gilles Serasset [3].

The Papillon schema redefines the DML entry element with the Papillon lexies and axies structures. It then contains the monolingual eixe structure common to all languages.

The language specific shemata define language specific elements like the list of parts-of-speech or the language levels.

The DML and Papillon XML schemata and XML examples are available online at http://www-clips.imag.fr/geta/services/dml/dml.xsd.

To write an XML entry, 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.

2 Detailed Example of a French Lexie

A lexie is an entry of a Papillon monolingual dictionary. The structure of the entries, that is microstructure of the monolingual dictionaries, is based on the structure used for the formal lexical database DiCo of the OLST laboratory at Universite de Montreal.

The encoding methodology is directly borrowed from the explanatory and combinatorial lexicology, which is part of the meaning-text theory elaborated by Igor Melc'uk and his colleagues.

The attribute id of the lexie element is an internal unique id. It is hidden from the users. If the lexie is deleted, its id remains. It can't be reused. It is metalinguistic information. The id is built by adding a . to the headword and then a unique integer.

The attribute basic is a boolean. It indicates if this lexie is the basic lexical unit of the vocable. The information isd taken from the DiCo database: if one vocable has more than one lexie, the basic lexie ie basic= true is a hidden lexie containing all the common elements of the vocable.

```
<lexie d:id="meurtre.1" basic="true">
```

The headword element is a common DML element declared in the DML schema. It represents the headword of the lexie. The hn attribute is the homograph number of the headword. It is not necessary linked to the lexie id.

```
    <headword hn="1">meurtre</headword>
```

The reading element is not useful for languages that not use kanjis. It is used in Japanese to write the headword in kanas.

```
<reading>meurtre</reading>
```

The pronunciation element is used to indicate the pronunciation of the headword. As several encodings can be used (GETA simplified encoding for French, romaji for Japanese, IPA, etc), the encding must be indicated.

```
<pronunciation encoding="GETA">meu+rtr(e)</pronunciation>
```

The part-of-speech element is used to indicate the part-of-speech of the lexie. The list of parts-of-speech for one language has to be described in language-specific schemata.

```
<pos>n.m.</pos>
```

The language-levels element is used to indicate the various language levels of the lexie. The list of language levels for one language has to be described in language-specific schemata.

```
<language-levels>
  <politeness/>
  <usage/>
  <reference/>
</language-levels>
```

The semantic formula comes from the meaning-text theory. It is a formal definition of the lexie. It begins with a semantic label which list should be closed and properly defined for each language. Then, if the lexie is a predicate, it indicates the composition of the arguments of the predicate.

```
<semantic-formula>
```

The element sem-label tags the semantic label of the formula.

```
  <sem-label>action de tuer</sem-label>: ~ PAR L'
```

The element actor tags an actor of the semantic formula.

```
<actor>
       <sem-label>individu</sem-label>
```

The element sem-variable tags a semantic variable of the formula.

```
       <sem-variable>X</sem-variable>
     </actor> DE L'<actor>
       <sem-label>individu</sem-label>
       <sem-variable>Y</sem-variable>
     </actor>
</semantic-formula>
```

The government pattern comes also from the meaning-text theory. It is the syntactical realization of the arguments of the predicate defined in the semantic formula.

```
<government-pattern>
```

There might be more than one government pattern (called "modifications") for the same lexical unit. The nb attribute of the mod elements encode this.

```
<mod nb="1">
  <actor>
    <sem-actant>X</sem-actant> =
       <synt-actant>I</synt-actant> =
       <surface-group>
      <surface>
        <reflexie xlink:href="#de">de</reflexie>
N</surface>,
        <surface>A-poss</surface>
    </surface-group>
  </actor>
  <actor>
    <sem-actant>Y</sem-actant> =
       <synt-actant>II</synt-actant> =
       <surface-group>
      <surface>
        <reflexie xlink:href="#de">de</reflexie> N
        </surface>,
        <surface>A-poss</surface>
    </surface-group>
  </actor>
</mod>
</government-pattern>
```

This lists the lexical functions and their results applicable to the lexie headword.

```
<lexical-functions>
  <function name="Qsyn">
    <valgroup>
      <value>
```

Reflexie reference to another lexie with an xlink href attribute. It is used to build a monolingual network.

```
        <reflexie xlink:href="#assassinat.1">assassinat</reflexie>
      </value>
    </valgroup>
  </function>
  <function name="S1">
    <valgroup>
      <comment> Nom pour X </comment>
      <value>
        <reflexie xlink:href="#auteur.1">auteur</reflexie> [de ART
~]</value>
    </valgroup>//
```

The lftype attribute is used to indicate if the sense of the lexical function is included in the lexical function lftype=’merged’ or not. It is equivalent to the // sign in the DiCo format.

```
      <valgroup lftype="merged">
         <value>
           <reflexie xlink:href="#meurtrier.2">meurtrier-n</reflexie>
         </value>
       </valgroup>
      </function>
    </lexical-functions>
```

Lists some usage examples of the headword of the lexie.

```
    <examples>
      <example d:id="e1">C'est ici que le double meurtre a √©t√©
commis.</example>
    </examples>
```

Lists some full idioms containing the headword of the lexie.

```
    <full-idioms>
      <idiom d:id="i1" xlink:href="papillon-axi.xml#axii0004">_appel
au meurtre_</idiom>
    </full-idioms>
```

The <more-info> element is used to store additional XML information for the lexie. It contains typically, information from recuperated resources. The content of the element is skipped for the validation. It just need to be valid XML.

```
    <more-info>
      <fem>information taken from the French-English-Malay
dictionary</fem>
    </more-info>
```

The <axies> element encode links between a lexie and various axies.

```
    <axies>
      <refaxie xlink:href="papillon-axi.xml#axil0001"/>
    </axies>
  </lexie>
```

3 Proposal Changes in the Structure

3.1 String encodings

Problem: For the moment, string litterals are encoded as simple strings. But when the user does not know how to read the language, s/he cannot read the strings. For example, it is ver difficult for someone not used to kanjis to read Japanese examples.

Proposal: All strings should be encoded as a text object with string+pronunciation. For Japanese, do we encode both romaji & yomigana ? If we encode all strings with objects, we can delete the <pronunciation> element for the headword.

```
<headword>
  <text><string> 殺人 </string><pron>satsujin</pron></text>
</headword>
```

3.2 The Reading element

Problem: The reading element is indicated only for the headword. For Japanese, do we add a reading element for all the strings?

Proposal: Leave it as is.

3.3 The Id attributes

Problem: The Id attributes and headword are written in capital letters in DiCo database. It has to be discussed if it is necessary to write them in capital letters knowing that it does not exist in Japanese.

Proposal: Encode the id attribute and the headword in small letters for roman scripts.

3.4 The basic attribute

```
<lexie d:id="meurtre.1" basic="true">
```

Problem: This basic was used to indicate that the lexie is the basic lexie of the vocable. This lexie is not a real one, it is used to store all the common information of the vocable to avoid duplication of the info. It had sense in the DiCo database but not in the Papillon dictionary because we duplicate the information.

Alternatives: 1. put of

2. implement a vocable object for the common informations of the vocables

Proposal: 1

3.5 The hn attribute

Problem: It is not the homograph number attribute, it is the number of the lexie !

Proposal: Call it lexienb.

3.6 The pos element

Problem: Now: conflict with a simple <pos> element and a <gram> element with a <pos> inside + more info. How to convert the DiCo Database ?

Alternatives: 1. map between DiCo pos and FeM pos, and put the other gram info into the other-info tag

2. enclose the <pos> tag into a gram tag and put the other info in the gram tag too.

Proposal: 1

3.7 Axie links

Problem: Now there are double links. Ones from lexies to axies and ones from axies to lexies. It is difficult to maintain the coherence.

Proposal: Keep only the links from axies to lexies and manage all the links from the axies dictionary. The Papillon application is already implemented to take this into account.

4 Conclusion

If everybody agree on te changes, we declare this structure as the version 1.0 of the Papillon common monolingual structure.

5 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. Gilles S rasset (1994) SUBLIM: un Syst me Universel de Bases Lexicales Multilingues et NADIA: sa sp cialisation aux bases lexicales interlingues par acceptions.Th se de nouveau doctorat, Sp cialit Informatique, Universit Joseph Fourier GRENOBLE 1, 194 p.