# The *Making of* a dictionary

***(Part 1.)*** *The "typographic and editorial" views vs. the "lexical" view*

Dictionaries written with standard word-processing software purely observe what we could call here a "typographic and editorial" approach: They mainly focus on the "final" representation of the lexicon, as it will eventually look in printed form – That is, for instance, with head-words set in bold face, usage hints and examples printed in italic, grammatical information and references rendered in smaller typeface than the rest of the text, etc.

> The *typographic view* is concerned with the two-dimensional printed page, including information about line and page breaks and other features of layout.
> The *editorial view* is the one-dimensional sequence of tokens which can be seen as the input to the typesetting process; the wording and punctuation of the text and the sequencing of items are visible in this view, but specifics of the typographic realization are not.

Of course, this may be perfectly fine for dictionaries aiming only at being published in printed form and presented to the public as a "nice book". The final printed representation, polished book-like form, is obviously what most compilers of dictionaries have in mind when they get at work, and we can easily see the main reason for that fact: it is often felt much easier to encode a dictionary in a form directly suitable for reading.

Electronic dictionaries, though, ought to be of a slightly different nature. Since the "encoded" source of the lexicon is made available to anyone and can be processed through various pieces of software, there is no reason to believe that one representation only of the information is possible. The main interest in having well-formed electronic dictionaries, in our opinion, is that other compilers and scholars should be able to almost automatically extract pieces of information from the initial document and present them differently – and even possibly use the same information to produce a wholly different work, not envisioned as a possibility by the initial compiler.

For this purpose to be achieved, though, the compiler first has to focus on the "lexical" view of the dictionary: that is, the way the underlying information structure is to be encoded, without concern for its exact textual representation. These issues are discussed in the *Text Encoding Initiative*'s Guidelines on which our Sindarin dictionary is based (TEI P4, §**12.5**), and it may be seen that a well thought lexical approach (while no doubt, in our experience, at the cost of extra efforts for the compiler!) eventually allows more flexibility to final users of the dictionary.

> The *lexical view* includes the underlying information represented in a dictionary, without concern for its exact textual form.

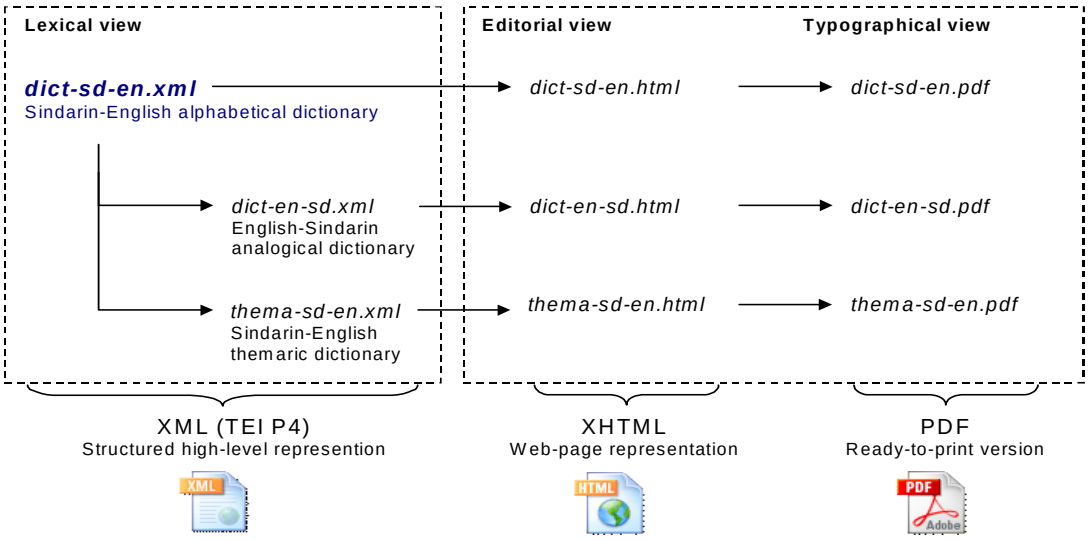
On the contrary, the "typographic and editorial"-only approach is very limited. Say, for example, that we have compiled an alphabetical dictionary including "domain" information such as "*Bot.*" for entries pertaining to botany, "*Geol.*" for those associated with geology, and so on. What if another compiler wants to produce a thematic dictionary, now sorting all the entries by domains, under separate headings for each? If the initial lexicon is only "typographic" or "editorial" (e.g. provided as an MS-Word or even an HTML document), it is pretty certain that the task will have to be performed almost manually (e.g. copying and pasting the entries accordingly into the new dictionary), because the very structure of the information is lost (or is, at least, mostly "visual").

Even when such a "typographic" dictionary is said to be "electronic" (in the sense that it is provided in some computerized encoding readable on a computer), one can say that it is so close to the printed lexicon that it does not actually differ from it — and therefore it suffers from the very same limitations: it is exactly the same result as when a printed dictionary is scanned and processed through optical character recognition (OCR), one cannot do much with it, except re-print it (or parts of it) more

or less exactly as it initially was. In the worst case, if the dictionary is provided in some ready-to-print format (e.g. as a PDF document), there is even less possibility to easily re-use the material differently...

But had the compiler adopted a purely lexical format in the first instance, that we would have been able to parse the document nearly automatically and to extract the "domain" information for further processing. When designing *Hiswelókë's Sindarin dictionary*, we had that aim at mind: not only to produce a "Sindarin dictionary" that Tolkien's fan would (possibly) enjoy, but also (mainly) a document written in such a way that it would allow other usages than the ones we were originally considering. We therefore selected an encoding format that could fit these requirements and tried to adopt a purely "lexical" approach from the start of the project. The *Text Encoding Initiative* specifically targeted dictionaries (TEI P4, §**12**), so the choice was quite obvious (even though XML was still emerging in 1999 when the project was initiated). Other considerations also played a role in the selected format, and we will perhaps discuss these issues later.

But for now, here is a simplified illustration of the software processings we can use to generate *three* different dictionaries, in various formats, from the *same* initial source:




One goes from the XML view to the XHTML view using standard W3C techniques (namely, XSLT style-sheets). As part of the Sindarin dictionary project, we wrote several specific XSLT style-sheets for the various tasks we wanted to achieved.

Finally, various techniques can be used to obtain the final presentation (PDF in the above illustration). For instance, one could go through XSL-FO (*Formatting Objects*) and then use the appropriate software (e.g. Apache FOP), but we may also note that many word-processing software (e.g. MS-Word or OpenOffice) can easily import XHTML documents, and then export them to PDF.

Anyhow, let us illustrate what we have in each "view" (For our "hardcore" readers that *will* have a look at the actual dictionary in XML, please note the examples below are slighty edited for the sake of simplification – Please refer to **Part 2** for a better description of the XML encoding used in *Hiswelókë's Sindarin dictionary*).

<table>
<tr>
<td>

```
<entry id='sd0344'>
  <form>
    <orth>annui</orth>
    <usg type='lang' norm='S' />
    <pron>"Annuj</pron>
  </form>
  <gramGrp>
    <pos>adj.</pos>
  </gramGrp>
  <sense>
    <trans lang='en'>
      <def>western</def>
    </trans>
  </sense>
  <note type='source'>
    SD/129-131
   </note>
</entry>
```

</td>
<td>

```
<p id="annui" class="sindict"> <span class="entry"><b>annui</b></span> <small><i>S.</i></small> [ɑnnuj] <small><i>adj.</i></small> western ◇ <small>SD/129-31</small></p>
```

The XHTML representation here includes basic typographic information (bold and italics, character size...) and elements of punctuation (e.g. the brackets around phonetics and square symbol before the references). Besides the phonetics is now encoded in Unicode.

But some of the original "structure" has been lost: There's no explicit way to see that "adj." Is a part of speech, that "western" is an English gloss, etc.

</td>
<td>

**annui** *S.* ['ɑnnuj] *adj.* western ◇ SD/129-31

The final representation (on screen or event better in print). Nothing to say — It's nice. But what if one now wants to list all adjectives ending in "–ui"?

Look at the XML encoding again. There, it would just be an easy matter of listing all <entry> elements having their first <orth> ending with "–ui" and their <pos> cointaining "adj." — Such extractions are pretty straightforward in XML, e.g. using XPath expressions and/or XSLT style-sheets.

</td>
</tr>
<tr>
<td>

```
<entry id='sd0579'>
  <form>
    <orth>avar</orth>
    <usg type='lang' norm='S'/>
    <pron>"AvAr</pron>
  </form>
  <form type='inflected'>
    <number>pl.</number>
    <orth>evair</orth>
    <usg type='lang' norm='S'/>
    <pron>"EvAjr</pron>
  </form>
  <gramGrp>
    <pos>n.</pos>
  </gramGrp>
  <sense n='1'>
    <trans lang='en'>
      <def>refuser</def>
    </trans>
  </sense>
  <sense n='2'>
    <usg type='dom'>Pop.</usg>
    <usg type='gram'>esppl.</usg>
    <trans lang='en'>
      <def>the Avari,
<index level1='Elf' lang='en'/>Elves
who refused (...) </def>
    </trans>
  </sense>
  <note type='source'>WJ/380,
VT/47:12</note>
  <note type='comment'
lang='en'>This plural name was
(...)</note>
</entry>
```

</td>
<td>

```
<p id="avar" class="sindict"> <span class="entry"><b>avar</b></span> <small><i>S.</i></small> ['ɑvar] <small><i>pl.</i></small> <span class="form"><b>evair</b></span> <small><i>S.</i></small> ['ɛvɑjr] <small><i>n.</i></small> <b>1.</b> refuser ○ <b>2.</b> <small><i>Pop.</i></small> <small><i>esp. in the pl.,</i></small> the Avari, Elves who refused the invitation of the Valar ◇ <small>WJ/380, VT/47:12</small> ◈ <small>This plural name was known to the loremasters, but went out of daily use at the time of the Exile</small></p>
```

The same remarks as above apply here. Note moreover that the definitions are nicely numbered ("1.", "2.") and separated with a circle symbol.
This is pretty (though hardly readable by a human in this form, admittedly).

However, we have lost the fact that "Pop." is a domain information, as well as the fact that "Elves" in the second definition was indexed (as "Elf", so that this entry could be returned if one was to list all words referring to Elvish tribes).

</td>
<td>

**avar** *S.* ['ɑvar] *pl.* evair *S.* ['ɛvɑjr] *n.* **1.** refuser ○ **2.** *Pop. esp. in the pl.,* the Avari, Elves who refused the invitation of the Valar ◇ WJ/380, VT/47:12 ◈ This plural name was known to the loremasters, but went out of daily use at the time of the Exile

The final representation (on screen or event better in print).

Still nice to read, but still "one-way"… What if one now wants to build an analogical English-Sindarin dictionary, with that word being listed under "Elf"?

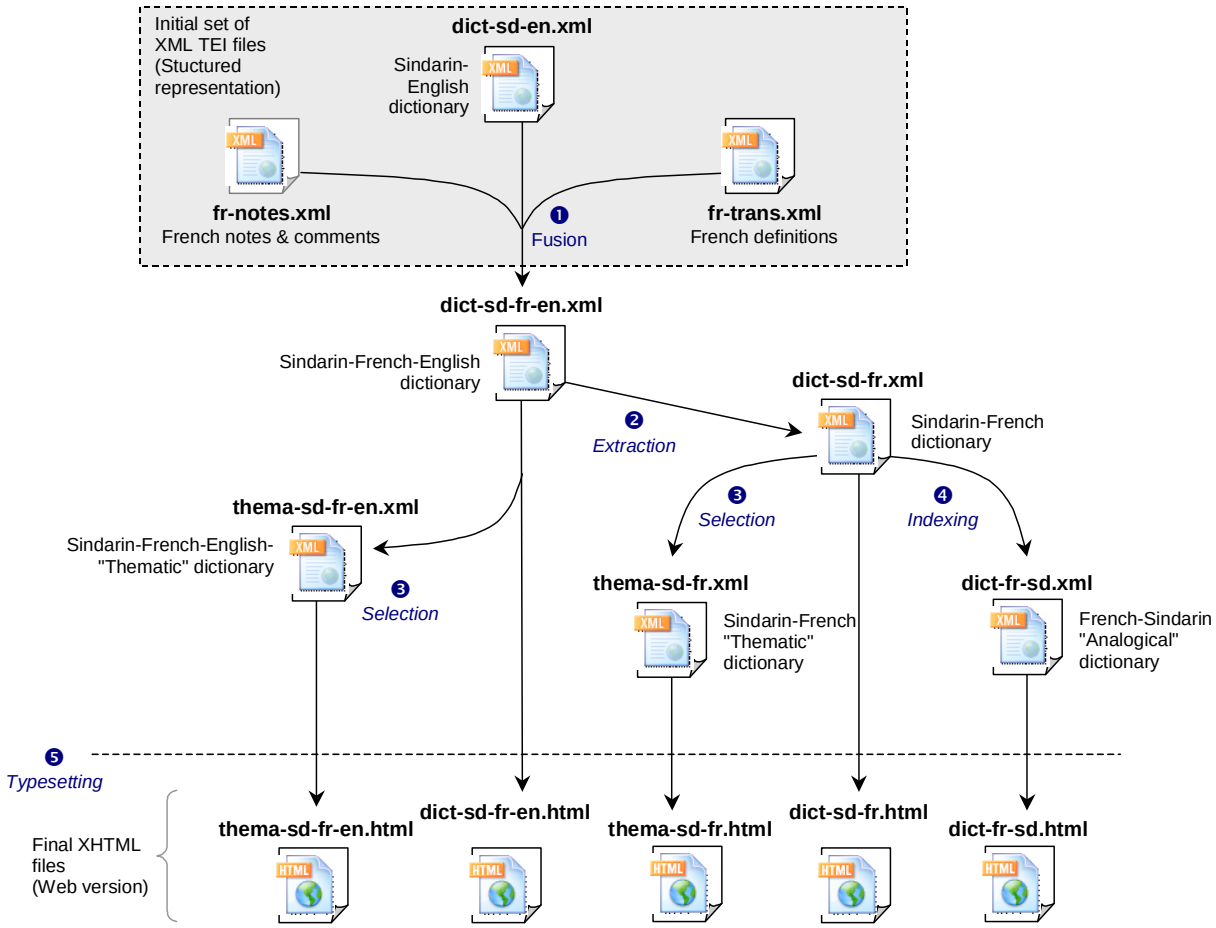
Look at the XML encoding again. There, it would just be an (somewhat) easy matter of collecting and sorting all disctinct <index> elements, and then listing all the entries where they appear. Such re-ordering is also feasible in XML, e.g. using XPath expressions and/or XSLT style-sheets (perhaps with a slight EXSLT flavor to ease assembling the distinct indexes, but almost all modern XSLT processors nowadays support these extensions to the W3C standard).

</td>
</tr>
</table>

Further above, we illustrated how three different dictionaries could be obtained from the same initial source. The process used to built foreign translations of the dictionary is nearly the same.

As before, the Sindarin-English “core” dictionary consists in a single file encoded in TEI P4 XML. Moreover, We now have two additional “translation” files (for definitions and editorial notes in the target language), both also encoded in XML…

And with these *three* files only, *five* dictionaries at least will be generated, all the steps being automated:




(**Step 1**) Definitions and notes are first merged with the main dictionary. The resulting file now is a tri-lingual dictionary in TEI P4 XML, for instance a Sindarin-French-English dictionary.

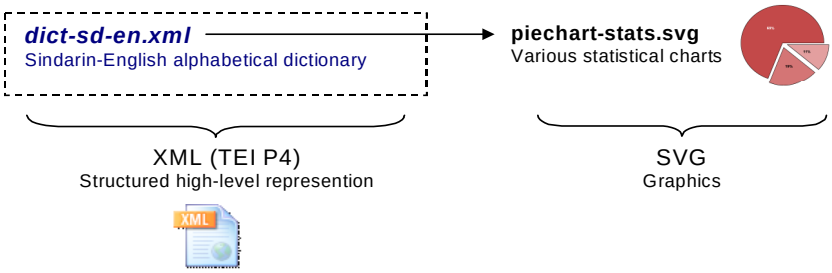
(**Step 2**) English definitions can be removed, for instance to generate a Sindarin-French dictionary.

(**Step 3**) Entries in these alphabetical dictionaries can be selected and sorted according to some pattern, e.g. by domain categories in order to produce thematic dictionaries.

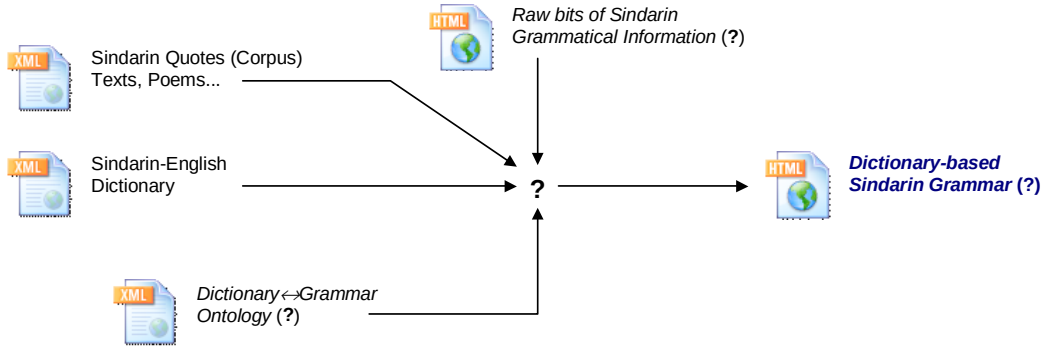
(**Step 4**) Likewise, an alphabetical dictionary can be indexed and re-ordered to generate an analogical (inversed) English-French dictionary.

(**Step 5**) Finally, all these dictionaries are converted to some format suitable for presentation and/or typesetting.

Of course, other applications of XML are possible. For instance, the lexical structured encoding also makes it feasible to easily count specific features of the underlying dictionary and to represent them in statistical charts:




And let's even dream a little (at the time writing this article)… A dictionary obviously contains some grammatical information. A grammar for the same language would be based on existing quotes (our corpus of texts, poems, sentences, etc.), would quote words from the dictonary and would also refer to the same grammatical categories as those used in the dictionary…




Needless to say, nevertheless, this is currently out of the scope of *Hiswelókë's Sindarin dictionary*. But if you have read that far, we hope, nevertheless, that this introductory article provides a clear vision of what we have tried to achieve and that you now better understand the choices we made.

**Part 2** will discuss strategies for encoding entries in *Hiswelókë's Sindarin dictionary*.

**Part 3** will list and document all files (XML, XSLT etc.) used to build the dictionary.