Khmer Lexicon Development

Chea Sok Huor, Top Rithy, Chhoeun Tola
csh007@gmail.com, toprithy@gmail.com, tola.ch2004@gmail.com

Abstract

The definition of a Text Base Management System is introduced in terms of software engineering, Goeser and Mergenthaler. *This gives a basis for discussing practical text administration, including questions on the experience in Khmer Lexicon Development in term of complex string objects and incorporated with related information. Moreover, there are some techniques to improve performance and manageability of data files will also be described in Methodology.*

1. Introduction

Since Khmer Unicode developed, many Software Departments have started to use Khmer Localization in their services. At the same time, some other organizations are also working with this significant project. Concern about the output of Localization Khmer language in Computer is not only dependent on technical factor but is also language and culture sensitive, that is why the development process needs to involve with both technicians and linguistic experts.

Consider some advanced language processing programs such as Machine Translation, POS Tagger, Semantic Web and Information Retrieval. They require a sufficient mechanism which is responsible for answering language information for a specific display as well as processing. Considering the concrete mechanism, Khmer lexicon project is much needed and was developed since phase 1 of the PLC (PAN Localization Cambodia) in order to satisfy the requirement for further project development.

The Khmer Lexicon is developed to aid a number of utilities within Khmer language, specific in vocabulary or rule base used for a particular professional application. According to these responsibilities, Khmer Lexicon provides many built-in methods and is categorized into two parts, PLC_LexiconUser and PLC_LexiconEditor. Within the output of Khmer lexicon project developers have possibility to filter information very deeply depending on the complexity of criteria that place into processing section.

2. Methods

2.1. Understanding in CHUON NATH dictionary

CHUON NATH dictionary is the official dictionary approved by the government and published by Institute BOUDDHIQUE, Phnom Penh, 1967. Up till now we have used the fifth edition of this dictionary and also had some changing of the previous information for each generation. The contents of the dictionary include all categories of word that are used in official presentation, technical documentation and idiom as well. Moreover, this dictionary includes many words that derive from Pali or Sanskrit, thus it usually useful for translate every document related with Buda religion or ancient documents.

Why CHUON NATH dictionary: In order to develop Lexicon we were required to get a huge number of data dimensions such as Part of Speech (POS), Root language, Alternative spelling, Synonym, Antonym, Hyponym, Phonetic etc... Thus there it was a time-consuming task for collecting data to place into each Lexicon dimensions.

Even though CHUON NATH dictionary is an official dictionary and reliable for validate document, it still has some lack of data dimensions with respect to Lexicon data requirement, because not more than 10% of the content in CHUON NARTH dictionary includes Antonym, Hyponym, Synonym data. Besides this missing data, this official dictionary provides a large number of information for Lexicon requirement; in addition other dictionary is not more reliable and detailed than CHUON NARTH dictionary in term of official reference and size.

Due to these reasons CHUON NATH dictionary was selected as the source of information for Lexicon data. Therefore, the architecture for Khmer Lexicon is designed to be compatible with the data structure of this paper based dictionary.

Even though CHUON NATH dictionary does not have enough information to complete the Khmer Lexicon data dimensions, it is still possible to extend data such as Synonym, Antonym, Hyponymy for the next version.

2.2. Architecture of the Khmer lexicon

The architecture of Lexicon API will be divided into two parts: **PLC_LexiconUser** and **PLC_LexiconEditor**, as can be seen in Figure 1. These last two parts are designed for manipulation and updating data. The details of each part will be described in the following section.

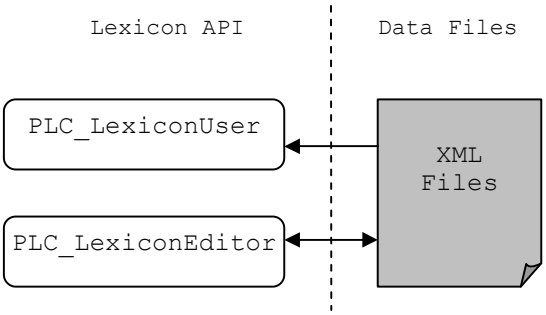

Figure 1: Khmer lexicon architecture

2.2.1. PLC_LexiconUser. PLC_LexiconUser is an API (Application Program Interface) specifically designed for manipulation or data searching purposes. Thus the functional specification of this part is robust performance.

For the PLC_LexiconUser function, Searching Information from Lexicon data (XML file format), it is necessary to show the structure of Lexicon Data Files, so we would like you to take a look at the way we store data in to hard drive and the way we read those data under any specific criteria.

2.2.1.1. The way we store data. There are a few steps in order to accomplish storing data. As mentioned above Lexicon Data is stored in XML (Extensible Markup Language) files, which is a W3C-recommended general purpose markup language for creating special purpose markup languages, capable of describing many different kinds of data, W3C. Lexicon is practically management of text and can simply be separated into management of Head-Word sets (as complex string objects and incorporated with related information). As a result, every Head-Word and its related information will be stored in an XML file. And those files will be stored differently in a specific File-Path based on the series of main consonants in the Head-Word.

For example: Head-Word = កាំ

កាំ = ក + ◌ា + ◌ំ + ◌ ◌ ◌

+Generating File-Name

ក = U-1780

◌ា = U-17B6

◌ំ = U-179A

◌ ◌ ◌ = U-17B8

Eliminate 1st byte from each Unicode characters
File-Name = 80-B6-9A-B8.XML

+Generating File-Path

Series of main consonants = ក, ◌ំ

File-Path = ..\Lexicon\ក\◌ំ

+As a result, Head-Word កាំ will be stored in ..\Lexicon\ក\◌ំ\80-B6-9A-B8.XML.

2.2.1.2 The way we read data. It is simply the reverse of the previous process (the way we write data). Firstly we get a Head-Word as a parameter to be searched. After that, we generate File-Name and File-Path and then check whether the target file exists or not. Finally, we use XMLDocument class (in System.XML name space) as xml parser to parse data in the target xml file.

Due to the data structure that was used in the CHUON NATH dictionary (paper based dictionary), we came up with a compatible XML Structure, and it was coded in DTD (Document Type Definition) format which contains information about the format of the XML document. A sample of this is as follows:

```
<!DOCTYPE WordEntity [
<!ELEMENT WordEntity (HeadWord ,
WordSenses?)>
<!ELEMENT HeadWord (#PCDATA)>
<!ELEMENT WordSenses (SubWordSense+)>
<!ELEMENT SubWordSense (Phonetic? ,
AlternativeSpell? , POS? , RootLanguage?
, Definition, Example?)>
<!ELEMENT Phonetic (#PCDATA)>
<!ELEMENT AlternativeSpell (#PCDATA)>
<!ELEMENT POS (POSName+)>
<!ELEMENT POSName (#PCDATA)>
<!ELEMENT RootLanguage (RootEntry+)>
<!ELEMENT RootEntry (RootName ,
RootDescription?)>
<!ELEMENT RootName (#PCDATA)>
<!ELEMENT RootDescription (#PCDATA)>
<!ELEMENT Definition (#PCDATA)>
<!ELEMENT Example (#PCDATA)>]>
```

2.2.2. PLC_LexiconEditor. PLC_LexiconEditor is an API specifically designed for editing and updating content of Lexicon Data Files (XML Files). As we have mentioned the above two solid processes: The way we read and write data to XML Files, we suppose there is no misunderstanding about the way we manage Data File and Directory Structure. PLC_LexiconEditor is an API, which functions as

data adapter and is responsible for validating data base on DTD and storing in to the target XML file.

2.3. Maintainability

With a huge size of Lexicon data, it is necessary to have a sufficient file structure, so we can manage and maintain easily. Otherwise, it will result in slow performance and corrupted data files. Due to the above issues, Lexicon Data was designed and stored in separated files for individual Head-Word. Directory Structure is manageable and easy to fix, in case files become corrupted. Samples of the file structure and directory structure are shown in Figures 2 and 3 respectively.

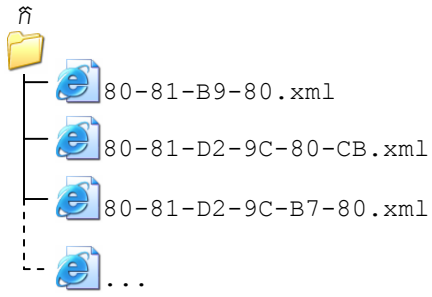

Figure 2: File structure

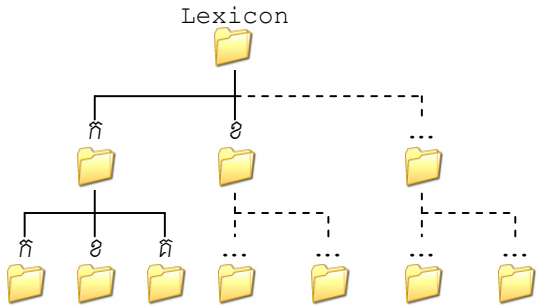

Figure 3: Directory Structure

2.4. Independent storage

Khmer Lexicon data file was formatted in XML format, which is recommended by W3C and the XML is one among other standard markup languages. Concerning about robust technology for data storage and maintenance, why didn't we choose a Relational Database System such as SQL Server, MySQL, Ms Access, etc.? There are some problems that will bother the user for further use, because this will require the user to have Database program installed before running Lexicon. This kind of thing will increase hardware and software requirement that will come from the Database program.

According to our experience with the previous version of Khmer Lexicon, using Database program (Microsoft Access) for data storage is much slower than storing data in separated xml files. Moreover, there is a risk, if Lexicon Data is stored in a single file (Microsoft Access file) in case the whole database file or even a small part is corrupted, it also causes the program to shut down.

3. Results

As a result, total number of Khmer Lexicon data files and folders is:

+XML files: 33,888 files (equivalent to number of Head-Words).

+Folders: 35,128 folders.

+Actual Size: 33.9MB

+Size on disk: 133MB

Searching performance: We have tested on a PC that has software and hardware capacity:

+Processor: Intel® Pentium® 4 CPU 2.40GHz

+Memory: 256 MB

+OS: Microsoft Windows XP Professional SP2

The result we are going to mention here is focused only on timed consumed during searching process. However, displaying process will take long or short time depending on what control or tool that is going to be used for displaying on screen, this process is excluded. The results are shown in Table 1.

Table 1: Search time

Criteria (Head-Word/Wildcard)	Time Consumed
ការី	5Miliseconds
កង្កែបអាចម៍គោ	7Miliseconds
អមោឃជីវិត	4Miliseconds
ល្ហែងលួងលោម	9Miliseconds
ពស្លែក	6Miliseconds
បុរាណសម័យ	7Miliseconds
រថភ្លើង	8Miliseconds
ការ*	38Miliseconds
ក*រ	187Miliseconds
ក*	3Sec 15MilSec

Due to the experiment result in Table 1, we notice that searching for a Head-Word (Fix string) consumed lesser than 10Miliseconds.

4. Conclusion

Khmer Lexicon lies at the head of further Language analysis research development. The task of acquiring a large-scale target language lexicon for further language processing base application can be daunting. We have shown that an efficient process for this acquisition may be developed by first determining the desired features of the process and then building efficient tools to facilitate different steps within the process. These tools have allowed us to make the acquisition process both manageable and effective, Leavitt, Lonsdale, Keck and Nyberg.

5. References

[1] CHUON NATH, Dictionnaire Cambodgien, Edition de L'Institut Bouddhique, Phnom Penh, 1967.

[2] W3C (World Wide Web Consortium), www.w3c.org.

[3] John R. R. Leavitt, Deryle W. Lonsdale, Kevin Keck, Eric H. Nyberg. Tooling the Lexicon Acquisition Process for Large-Scale KBMT, http://www.lti.cs.cmu.edu/Research/Kant/PDF/take3.pdf.

[4] S. Goeser, E. Mergenthaler. TBMS: Domain Specific Text Management and Lexicon Development, http://acl.eldoc.ub.rug.nl/mirror/C/C86/C86-1056.pdf .