Encoding Conversion Utility for Khmer

Chea Sok Huor, Atif Gulzar, Ros Pich Hemy, Neak Longchrea
PAN Localization Team, Cambodia
csh007@gmail.com, atif.gulzar@gmail.com, pichhemy@gmail.com, longchrea@gmail.com

Abstract

This paper discusses the research on the encoding conversion from Khmer non-Unicode fonts such as Limon, KHEK etc. into Unicode. The document is divided into three main sections. Section 2 discusses the analysis of the non-Unicode font representation and the conversion technique. Section 3 presents the result of the studies. Section 4 discusses the findings and implications of the work that will present all the cases that have not been handled yet.

1. Introduction

According to the Unicode Standard Version 4.0 from the Unicode Consortium Khmer script, called *aksaa khmae* ("Khmer Letters"), is the official script in Cambodia. It is descended from the Brahmi script of South India, as are Thai, Lao, Myanmar, Old Mon, and others. The exact sources have not been determined, but there is a great similarity between the earliest inscriptions in the region and the Pallawa script of the Coromandel Coast of India. Khmer has been a unique and independent script for more than 1,400 years.

Before the creation of Khmer Unicode, Khmer used Latin code to represents its characters. Many fonts are being created to facilitate the office task. Consequently, there is no standard for representation. A character can be represented in two different codes for two different fonts. As a result, data transfer across fonts is impossible. This leads to the invention of Khmer Unicode and the development of a tool to convert from those non-Unicode font documents to Unicode format, which is an indispensable task.

Grammatically, Khmer syllable must respect the spelling order. Since the non-Unicode fonts input method is very flexible, users can combine whatever they want without respecting the meaning and spelling orders, and the task of the conversion is a challenge task. The rest of this paper will present a detailed technique for the conversion and the results of the research.

2. Method

2.1. Architecture of the application

To ensure the user-friendliness, the embedded applications for MS Office are proposed. The system will be developed by dividing its functionalities into three main modules: MS Office Suite, Automation Application, and Conversion Assembly. Figure A.1 of Appendix A illustrates the system.

- **MS Office suite:** is the existing MS Office suite (Word, Excel, FrontPage, Power Point, Outlook and Publisher).
- **Automation application:** All the applications in this module act as the intermediate between the conversion assembly and MS Office suite. Its tasks are to get the content of the document to be converted from the MS Office suite into conversion assembly, to replace the converted text to MS Office suite, and to keep the format of the converted document the same as the previous one.
- **Conversion Assembly** is the core module of the system. Its main function is to convert a non-Unicode text into Unicode format. The rest of this paper will discuss this module, as it is an important part.

2.2. Problem in Khmer non-Unicode writing technique

Since there is no standard to assign the character among Khmer non-Unicode fonts, many problems occur for the representation.

- Some words can be written in different ways. For example, the word **ស៊ី** (SI) can be represented as the sequence of **ស** (SA), ុ (SRAK U) and ី (SRAK II) and also the

sequence of សំ (SA), ី (SRAK II) and ុ (SRAK U).

- Some commonly used words or syllables are represented in one code. For example, in Limon and ABC font families, the word ខ្ញុំ (KNYOM) is assigned to one code.
- A character can be written by combining many characters. For instance, In Limon and ABC fonts the character ញ (NYO) can be written as a sequence of ព (PO) ា (SRAK A) ្ញ (COEUNG NYO).

2.3. Khmer non-Unicode fonts script presentation analysis

The representation of Khmer characters in Khmer non-Unicode font is categorized into 7 main groups: consonant, subscript, dependent vowel, consonant shifter, various sign, independent script, and special script

2.3.1. Consonant

It is the main character for every of Khmer syllable. The consonant can be divided into three main groups, the consonants that can be used with MUUSIKATOAN such as ង ញ ន ម យ រ ល វ ប, the consonants that can be used with TRIISAP such as **ស ហ អ ប.** The rest cannot be used with both consonant shifters.

2.3.2. Subscript (also, called *coeng* (literally, "foot, leg"))

They are found on the left, right or bottom of the main consonant.

Khmer non-Unicode subscript can be divided into 3 categories according to its rendering position related to consonant as illustrated in the figure below.

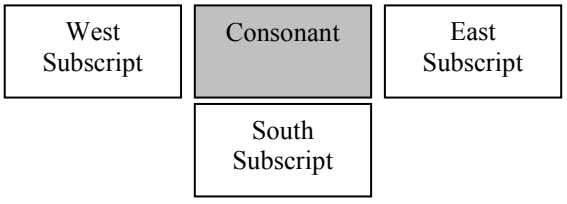

Figure 1: Khmer non-Unicode subscript rendering

2.3.3. Dependent vowels

Dependent vowels usually follow main consonant or subscripts because it cannot stand alone in Khmer writing. Khmer non-Unicode dependent vowel can be divided into 4 categories according to its position and the input sequence in non-Unicode font.

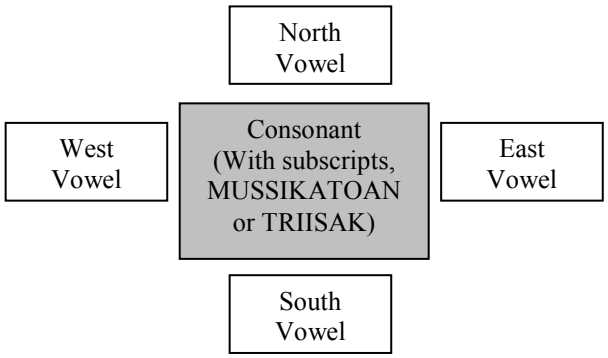

Figure 2: Khmer non-Unicode dependent vowel rendering

Problems with east vowels: The vowels that are visibly positioned on the east side of the base consonant are called east consonants. Since the non-Unicode font unable user to type a west-east vowel in one key stroke, its combinations are used. For example, in order to get the vowel "ែា", non-Unicode font user has to type parts of this vowel: "ែ" and the incomplete character AU.

2.3.4. Consonant shifters

Khmer consonant characters and signs are basically divided into two sounds –a and –o in Khmer language. Consonant shifter signs are used to shift between the two sounds. There are two consonant shifters: ◌ (KHMER SIGN MUUKSIKATOAN) and ◌ (KHMER SIGN TRIISAP).

- KHMER SIGN MUUKSIKATOAN ◌ shifts sound of a consonant character or sign from the second from –o to –a.
- KHMER SIGN TRIISAP ◌ shifts sound of a consonant character or sign from the second from –a to –o

2.3.5. Independent scripts

In Khmer non-Unicode font, the characters in this category refer to those characters that can be replaced or converted directly regardless of the position, order of combination rule. This category comprised of Independent vowels, currency symbol, digit etc…

2.3.6. Various sign

Various signs are always put at the end of the orthographic syllable in Khmer language.

2.3.7. Special script

The type refers to Khmer non-Unicode characters or words that do not exist in the Khmer Unicode range. For examples: the word ខ្ញុំ, in Limon or ABC fonts, is only one code and one keystroke. However, it should be written sequence of ខ ◌្ញ ◌ុ and ◌ំ.

Remarks: Since the writing rule in non-Unicode is not strict, some words or character can be typed or represented in many different ways. For example, ញ្ញ is a variation of ញ្ when it is used with subscript. In ABC and Limon font writing behavior, it is a combination of ញ + ◌្ញ.

Another example is the case of "incomplete AU". There is no Unicode represented for this character because it is part of Khmer dependent vowel ◌ៅ (SRAK AU).

2.4. Conversion algorithm

The conversion process is decomposed into two main tasks independently:

1. First, the input non-Unicode sentence is converted to Unicode sentence, but it remains in the same order as the non-Unicode sentence. This sentence is called Common Script Sentence (CSS). In order to enable the conversion of different fonts, an engine is generated for each font family. The main task of each engine is to translate the non-Unicode character into corresponding Unicode character.
2. Then, the conversion assembly reorders the sentence or CSS according to Khmer spelling order.

For example:

Input sentence: គេប្រៀង

The sequence of the sentence is

◌	គ	◌	◌	ច	◌	ង

First, the sentence is converted into Common Script Sentence (CSS) as the following picture.

◌	គ	◌	ច	◌	ង

Finally, the CSS is reordered according to Khmer spelling order as the following picture.

គ	◌	◌	ច	◌	ង

The problem of reordering the sentence into Khmer spelling order is a complex task and will be discussed in the rest of this section.

2.4.1. CSS reordering

Since there is no marker between Khmer words, the main process of the CSS reordering is to first decompose the CSS into different inseparable units and then reorder each unit according to spelling order. In Khmer language, Orthographic syllable is the main component (inseparable unit) in writing. The rule for forming the unit is precise. It can be a block of consonant, consonant shifter and subscripts surrounded by a block of vowel and various signs. The structure of the orthographic syllable combination is illustrated in the Figure 3.

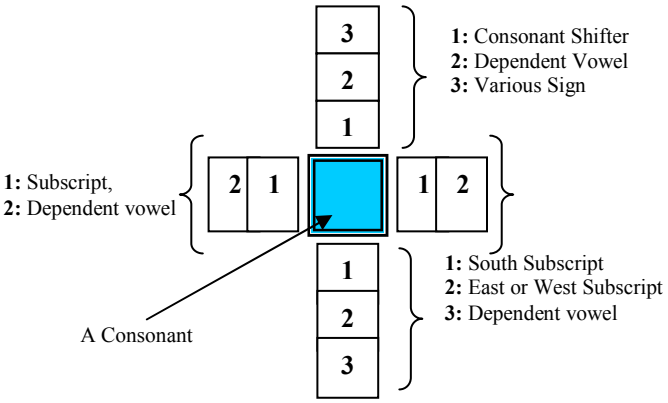

Figure 3: Structure of the orthographic syllable combination

The rule for ordering the orthographic syllable in Khmer non-Unicode writing is given as follows in terms of types of character:

1. Consonant
2. First subscript
3. Second subscript

4. Consonant shifter
5. First vowel
6. Second vowel
7. Third vowel
8. Various sign

Subscript priority

- South subscript
- East subscript
- West subscript

Vowel priority

- North vowel or east vowel
- West vowel
- South vowel

2.4.2. Orthographic syllable detection

The major problem for the CSS reordering is how to detect the orthographic syllable. Our main idea to solve the problem is to detect the transition state for each character of an input string. If there is no possible transition for any character, it means that the end of syllable is reached. Therefore, there is a need to analyze all the possible input in non-Unicode font writing behavior and create a possible transition lookup table as shown in Table 1 (also see Table 2).

Table 1: Possible transition lookup table

	C	CS	WV	NV	SV	EV	WSS	SSS	ESS	VS
C	0	1	0	1	1	1	0	1	1	1
CS	0	0	0	1	1	1	0	1	1	0
WV	1	0	0	0	0	0	1	0	0	0
NV	0	1	0	0	1	0	0	1	0	0
SV	0	1	0	1	0	1	0	1	1	1
EV	0	1	0	1	0	0	0	0	0	1
WSS	1	0	0	0	0	0	0	0	0	0
SSS	0	1	0	1	1	1	0	0	0	1
ESS	0	1	0	1	1	1	0	0	0	1
VS	0	0	0	0	1	0	0	1	0	0

Table 2: Abbreviation Chart (Initial states are WV, WSS and EV)

Shortcut	Abbreviation
C	Consonant
CS	Consonant Shifter
WV	West Vowel
NV	North Vowel
SV	South Vowel
EV	East Vowel
WSS	West Subscript
SSS	South Subscript
ESS	East Subscript
VS	Various Sign

Note: In the possible transition lookup table:

- Zero means ***NOT Possible transition*** and one means ***possible transition.***
- The current state is in ROW.
- The transition state is in COLUMN.

Example of the CSS reordering process:

Input string: សាលាក្តី

The scenario:

- First of all, **ស** is kept as current state
- The next character is ◌ា. According to the table above, there is possible transition from **ស** to ◌ា. Therefore, ◌ា is kept as the current state and the method goes to the next character.
- The next character is the consonant **ល**. There is no possible transition from the current state (◌ា) to the consonant **ល**. It means that the terminator is reached and the sequence of **ស** and ◌ា is the orthographic syllable.
- The process is restarted from the character **ល** by doing the same process as above.

2.4.3. Font dictionary concept

To ensure the extensibility and optimization, all the codes of the non-Unicode and its Unicode correspondent of each font family are stored in a file called the dictionary file. When the conversion assembly is loaded, the data in the dictionary are read and stored temporarily in the application. Therefore, many files are created according to the number of font families.

The structure of the files is as follows:

- Each line of the file represents one code of the Khmer script.
- A tab separates each section of the line.
- A line, which starts with #, is a comment line. If the assembly sees this in front of the line, it will directly skip it. Therefore, the user can add as many comments as needed.

Data line of the dictionary is represented as shown in Figure 3.

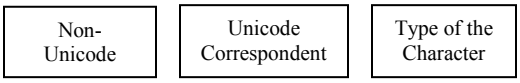

Figure 3: Dictionary data line representation

- **Non-Unicode code:** this section is to store the code of the non-Unicode font in hexadecimal.
- **Unicode code:** this section is to represent the code, hexadecimal format, corresponding to the non-Unicode code of the same line.
- **Types of the character:** In the application, the Khmer letter is categorized into 14 types:
  1. Normal consonant
  2. Special script
  3. Independent script
  4. Consonant shifter
  5. West subscript
  6. East subscript
  7. South subscript
  8. West vowel
  9. East vowel
  10. South vowel
  11. North vowel
  12. Various sign
  13. Consonant that can be used with MUUSIKATOAN
  14. Consonant that can be used with TRIISAP

2.5. Problems and solutions

2.5.1. Special scripts problem

Some characters in non-Unicode font cannot be mapped to Khmer Unicode. For example, ខ្ញុំ in Limon or ABC fonts is only one code (one keystroke). In fact, in Unicode, it is represented by ខ + ◌ួ + ◌ុ + ◌ំ.

The problem is handled by giving each special script a code, which is not redundant with other Khmer Unicode range and other Unicode.

2.5.2. Some special case problems

Since the rule in the non-Unicode is not strict, some word or character can be typed and represented in many different ways. Therefore, it is a big problem for the conversion tasks. As an example: ញ is a variation of ញ when it is used with subscript. In ABC and Limon font, it is a combination of ព + ◌ា. Another example is the case of incomplete vowel sign as mentioned earlier in this paper. This kind of character does not exist in Unicode. Therefore, if it is found alone, the system will convert it directly to complete character.

The problem is solved in the engine of each font since different problems occur in different fonts.

2.5.3. Some special case problems

Consonant shifters shift the base consonant between registers. Many problems are found concerning with consonant shifter.

- First, in Khmer language, the representation of scripts is not totally the same as its spelling order. For example, the word ស៊ី (SI) can be represented as the sequence of ស (SA), ◌ុ (SRAK U) and ◌ី (SRAK II) and the sequence of ស (SA), ◌ី (SRAK II) and ◌ុ (SRAK U). The correct sequence is ស (SA), ◌៊ (TRIISAP) and ◌ី (SRAK II).
- Second, not only the register reflects the base consonant, it also reflects the subscript.
- Finally, in Unicode order, if user visibly wants to keep the shifter in the script, the "ZERO WIDTH NON JOINER" must be added. For example ប៊ី = ប+◌៊+ ZERO WIDTH NON JOINER + ◌ី

In order to solve the problem, we need a careful study on the usage of Khmer consonant shifter. The steps to solve the problem are to:

1. Identify the consonants that can be used with TRIISAP and the consonants that can be used with MUUSIKATOAN.
2. Create a general rule for the order of scripts with respect to the Khmer script grammar.
   - **The priority of the register reflection:** As described in the problem above, not only the consonant that can be reflected by the register, the subscripts also. Hence, the main problem is to decide which one is affected by the consonant shifter if the syllable contains both consonant and subscript. According to our studies, the most common priority order is second subscript, first subscript, consonant that can be affected by MUUSIKATOAN and TRIISAP.

- **Zero width non-joiner detection**: The decision for inserting a ZWNJ is to look in the combination buffer if it contains a consonant shifter, consonant or subscript that can be affected by register and a north vowel. There is exception for the north vowel SRAK_AM. There is a need to consider if the east vowel is SRAK_AA or not. If east vowel is SRAK AA, then ZWNJ is added.
- **Detect a consonant shifter from a non-Unicode combination:** As showed in the first problem, the system has to detect the consonant shifter during the conversion. The solution to the problem is to look in the combination buffer if there is consonant or subscript and north vowel when we meet SRAK_U. If it completes the condition, it means there is consonant shifter in the Unicode combination.

3. Experiment Result

In our experiments, we tested the system performance of the approach with different non-Unicode font documents. The total size is around 5MB. The approach has achieved a rate over 95 percent. Mostly, the case that had not been converted correctly is not accepted semantically and visibly in Khmer script and grammar.

4. Discussion

4.1. The case of consonant ឡ (LA)

Some non-Unicode users type and represent this letter by typing ទ (TO) and ្ឡ (Subscript of BA). The visibility of the sequence is different from the correct character. Therefore, the sequence is not converted to the character ឡ, but it remains the sequence of ទ (TO) and ្ឡ (Subscript of BA).

4.2. SRAK AA_M ាំ and consonant shifter

The problem can be illustrated with an example of the word សុាំ. In Khmer spelling order, this word is the combination of **ស, ◌ុ, ◌ា** and **◌ំ.** However, in Khmer non-Unicode font, the user can represent it in three ways:

1. ស + ◌ុ+ ◌ា + ◌ំ → សុាំ
2. ស + ◌ា + ◌ំ + ◌ុ → សុាំ
3. ស + ◌ា + ◌ុ + ◌ំ → សុាំ

The system does not convert the third case to the sequence of **ស, ◌ុ, ◌ា** and **◌ំ**, but to the sequence of **ស ◌ា ◌ុ ◌ំ**. The reason is that the system uses graph to determine the terminator. Therefore, providing the ability to handle the case might affect many other cases.

4.3. Consonant NYO "ញ"

As described in the second section, non-Unicode users can represent the letter NYO in many different ways:

1. ញ
2. ព + ◌ា + ◌ូ → ញ
3. ព + ◌ា → ពា
4. ព + ◌ូ + ◌ា → ញា

The system does not convert the fourth case to the character ញ, but to the sequence of **ព ◌ូ ◌ា** due to its different representation from the correct character ញ.

4.4. Subscript of TA "្ត" and DA "្ដ"

In Khmer script, the subscript of TA and DA are visibly the same. Hence, the Khmer non-Unicode fonts represent it as one code. It is very difficult for the conversion assembly to identify which one the user means to write. The conversion does not handle the case and convert it to the subscript of DA.

4.5. Alternative cases

The problem can be demonstrated using the following example.

In Khmer non-Unicode font, the visibility of the sequence of is គ and then ◌ោ is as it is typed. After converting to Unicode, the system will give the sequence គ and ◌ោ. The rendering engine will adjust the vowel before the consonant to ◌ោគ, which is visibly different from its previous shape.

5. Conclusion

This paper presents an approach to convert from Khmer non-Unicode font strings into Unicode format. The approach has achieved a satisfactory result. The utility is the main bridge to increase the use of Khmer Unicode. It enhances the standardization and localization of Khmer scripts in the computer. Moreover, the project enhances the correct use of Khmer linguistic (the writing system).

In addition, the flexible design of the system ensures the extensibility and reusability. For example, recently FK font family, which is used by Ministry of Finance, is added to the system without major effort and time.

However, the speed of the conversion is not yet as high as expected, but acceptable. It is recommended to use the text file conversion if the document contains only text with the same font family and the same format.

5. Future Works

For the project, some work need to be done to improve the performance of the system. There is a need to:

- Find solution for some unhandled problems as discussed in the discussion part such as the subscript of TA and DA.
- Make the conversion applicable for more fonts.
- Convert data in the database and other applications other than Ms. Office application.

6. References

[1] C. Nath, *Dictionnaire Cambodgien*, Edition de L' Institut Bouddhique, Phnom Penh, 1967.

[2] http://ftp.unicode.org/charts/PDF/Unicode-4.0/U40-1780.pdf

Appendix A

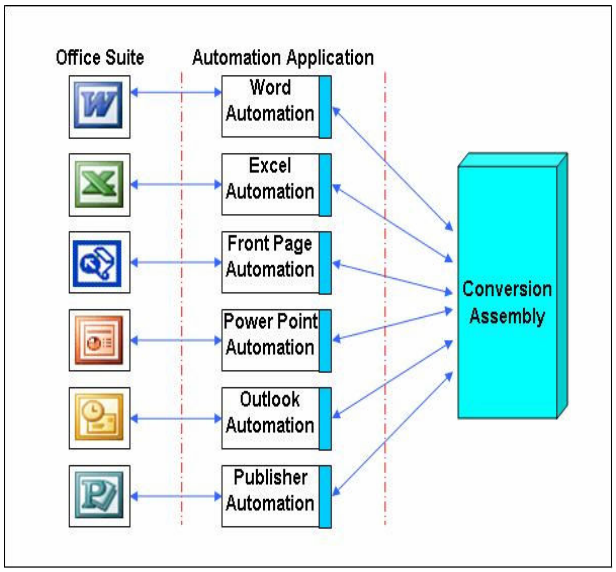

Figure A.1: Unicode Conversion Architecture