Two problems in Thai rendering

Eric Muller, Adobe Systems Inc.
July 18, 2004

1. A Thai refresher

The Thai script uses non-spacing combining marks for a number of functions:

- vowels above, e.g. U+0E34 ◌ิ THAI CHARACTER SARA I
- vowels below, e.g. U+0E39 ◌ุ THAI CHARACTER SARA UU
- tone marks above, e.g. U+0E48 ◌่ THAI CHARACTER MAI EK
- other signs above, e.g. U+0E4D ◌ํ THAI CHARACTER NIKHAHIT (which has the annotation “final nasal”)

When the Thai language is written, at most one vowel/other sign is used, and at most one tone mark is used. If only one mark above is used, it is placed just above the consonant. If two marks above are used, the tone mark is always above the vowel/other sign; in other words, the tone mark moves up to make space for a vowel or other sign.

For example:

2. Problem 1: typographic interaction

The combining marks have fixed combining classes. Since Unicode 3.0, the tone marks have a combining class of 107, the vowels and other signs above have a combining class of 0, and the vowels below have a combining class of 103 or 9 (in Unicode 2.0, each character had its own fixed class, ranging from 98 to 112, plus 128).

This choice of fixed classes – rather than the usual classes of 220 (below) and 230 (above) – reflects the use of the script in writing the Thai language. It is also very unlikely that another orthography using the Thai script would use these marks differently.

One of the functions of combining classes is to determine the rendering of sequences consisting of multiple combining marks. The underlying principle is expressed in section 3.11 of TUS 4.0, at the bottom of page 83: “Characters have the same class if they interact typographically, and different classes if they do not.” The typographic interaction is then regulated by the text in the rest of the section, and in particular on page 82: “Combining marks with the same combining class are generally positioned graphically outward from the base character they modify.”

Since the marks we are discussing here fall in different combining classes, the machinery of combining classes does not tell us how sequences such as <vowel above, tone mark> or <tone mark, vowel above> should be rendered. In fact, the example above is only one possible rendering of the string: there is nothing in the standard which indicates that the mai ek should be displayed above the nikhahit, or put another way, nothing that prevents the rendering from showing them in exchanged

positions. Therefore, to reliably interchange such text, some convention currently outside the standard has to be invoked. This is clearly not desirable.

One possibility would be to make the general category of the combining marks above 230, and that of the combining marks below 220. The existing words of the standard would then tell us the rendering of those strings. We believe this would not affect the canonical representation of *existing* data, but it would change the canonical representation of some string, thereby breaking the guarantee of stability of normalization.

The proposal is to clarify these situations by adding the following text to the Thai block description:

> For the purpose of rendering, the combining marks above (0E31, 0E34..0E37, 0E47..0E4E) are displayed outward from the base character they modify. In particular, a sequence containing <U+0E48 ◌่ THAI CHARACTER MAI EK, U+0E4D ◌ THAI CHARACTER NIKHAHIT> should be displayed with the nikhahit above the mai ek, and a sequence containing <U+0E4D ◌ THAI CHARACTER NIKHAHIT, U+0E48 ◌่ THAI CHARACTER MAI EK> should be displayed with the mai ek above the nikhahit. Similarly, the combining marks below (0E38..0E3A) should be displayed outward from the base character they modify.

This proposal does not prevent input processors from helping the user by pointing out or correcting typing mistakes, may be taking into account the language. For example, since the string <mai ek, nikhahit> is not useful for the Thai language and is likely a typing mistake, such an input processor could correct it to <nikhahit, mai ek>.

3. Problem 2: Sara am

The Thai block also contains the character U+0E33 ำ THAI CHARACTER SARA AM. This character is unusual in that it represents both a non-spacing mark on the previous base character, and a spacing mark on its own. In fact, it has a compatibility decomposition of <U+0E4D ◌ THAI CHARACTER NIKHAHIT, U+0E32 า THAI CHARACTER SARA AA>.

This character was presumably encoded for compatibility with the TIS 620-2533 standard.

It is likely that most existing data use the sequence <sara am> rather than the sequence <nikhahit, sara aa>, because sara am is available as a single keystroke on most keyboard layouts for Thai (starting with TIS 820-2538, and continuing with the keyboard drivers provided with Windows and MacOS).

What is not clear from the Unicode standard is whether text which involves both a nikhahit and a tone mark followed by a sara aa, can be represented by the string <..., U+0E48 ◌่ THAI CHARACTER MAI EK, U+0E33 ำ THAI CHARACTER SARA AM>.

Most implementations seem to support this representation, and it is probably the most common form in actual data. Therefore, we would like to clarify the standard.

The proposal is add the following text in the Thai block description:

> When the character U+0E33 ำ THAI CHARACTER SARA AM follows one or more tone marks (U+0E48 .. U+0E4B), the nikhahit that is part of the sara am should be displayed below those tone marks.

Document History

Author: Eric Muller

Revision	Date	Comments
1	July 18, 2004	Initial version