Tamil Encoding in Unicode - A Comparative Study

P.Chellappan
Palaniappan Bros., 14, Peters Road, Chennai - 600014
<Email: chellappan@vsnl.com>

INTRODUCTION

During the TamiNet99 Conference, which was held in Chennai, several papers were presented regarding the need to change the present Encoding of the Tamil Script in Unicode that occupies the Unicode locations U+0B80 to U+0BFF. Among the people who wanted a change, there were two schools of thought. There was one group that wanted assignment of unique locations only for the Uyir, Ayutham, Mei and Grantha characters (12+1+18+5+1). The other group wanted allocation of space for all the 313 Tamil characters. The author of this paper had also presented a paper calling for an encoding scheme that has a unique allocation for each of the 313 Tamil characters (Uyir, Mei and Uyir-Mei including the Grantha characters) and also for other Tamil Symbols. The purpose of this paper is to make a comparative study of the three different encoding schemes, so that a decision can be taken at the earliest.

THREE SCHEMES

1) THE PRESENT UNICODE SCHEME

This scheme allocates unique locations for the Uyir, Akaram Eriya Mei, Vowel Modifier and Symbol. Instead of treating an Uyir-Mei character as a combination of a Mei character and an Uyir character, it is treated as a combination of an Akaram Eriya Mei character and a Vowel modifier character. It also treats the Tamil Grantha characters 'ksha' and 'sri' as conjunct consonants. In order to be compatible to the other Indic scripts, the allocation of these characters are not as per the Tamil sort order. 128 locations are sufficient for this scheme.

Advantages:
All the Indic languages are allocated a block of 128 locations each and similar characters occupy the same relative location within the block. This enables easy transliteration possible between the Indic languages. Just a relative shift of locations would be sufficient for transliteration from one Indic language to another.

It helps in Natural Language Processing, Spell Check etc since the Uyir-Meis are already split into its basic components.

Disadvantages:
Since the Uyir-Meis are represented as combinations of an Akaram Eriya Mei and a Vowel Modifier each one of these characters would take up 32 bits (16 bits each for the Akaram Eriya Mei and Vowel Modifier characters). This results in large file sizes and also poor efficiency in processing.

Because of the same reason, there is no 1:1 relationship between characters and glyphs. Hence Glyph substitution will be required for proper display rendering. Tamil cannot be implemented in Level 1 of Unicode like English and the CJK (Chinese, Japanese and Korean) languages.

It ignores the natural sort order of the Tamil script. Hence it requires a separate Weight Table for proper sorting.

Only softwares that are Tamil enabled can be used.

This scheme does not follow the proper Tamil Grammatical rules.

2) PROPOSAL 1

In this scheme unique locations are allotted only for the Uyir, Mei, Grantha and symbol characters. The proper grammatical structure of Tamil is implemented in this scheme. All Uyir-Mei characters are represented as combinations of a Mei and a Uyir character. 128 locations are sufficient.

Advantages:
It helps in Natural Language Processing, Spell Check etc since the Uyir-Meis are already split into its basic components.

It maintains the Grammatical Structure of the Tamil language.

Since the proper sort order is maintained while allocation itself, straightforward sorting, without the need for a separate sort table, is possible.

Disadvantages:
File sizes are large since the Uyir-Meis are treated as combination character of Mei and Uyir character. This in turn leads to poor efficiency.

Since there is no 1:1 relationship between Characters and Glyphs, Level 1 implementation of Unicode is not possible.

Only softwares that are Tamil enabled can be used.

Transliteration to other Indic languages is slightly more difficult than the existing scheme.

3) PROPOSAL 2

This proposal envisages allocation of unique locations for each of the 313 Tamil characters and all the required Tamil Symbols. In this scheme all Uyir, Mei and Uyir-Mei characters including the Grantha characters are represented as single 16 bit characters (Unicode Characters) and not as combinations of Mei and Uyir characters. This proposal will require increase of the number of locations assigned for the Tamil script from 128 to 313+.

Advantages:
Since all the characters are represented only as 16 bit characters, the file sizes are smaller and as a result it is more efficient.

There is a perfect 1:1 relationship between characters and glyphs. Hence Tamil can be implemented even in a software that is Unicode Level 1 compliant. Literally all available softwares can be used for Tamil, without difficulty.

Sorting is easy and there is no need for separate Sort Weight Tables.

Disadvantages:
Since all Uyir-Meis are stored as single characters, one will have to use a mathematical manipulation to split it into its Mei and Uyir component. Hence at a first glance one will be led to believe that it is not suited for Natural Language Processing, Spell Check etc., But since efficiency is lost only in a memory operation as opposed to the loss of efficiency in a storage device read/write operation, this scheme still results in a better performance than the first two schemes.

Transliteration to other Indic languages is slightly more difficult than the existing scheme.

TESTING

The above comparison of the three schemes clearly shows that the all character representation scheme (Proposal 3) is the best. However all the theoretical discussions will have to be verified by proper testing.

Since both the Existing Scheme and Proposal 1 encode the Uyir-Mei characters as combination characters, efficiency of both these schemes would be similar except maybe in Natural Language Processing, Spell checking etc., where Proposal 1 could be better.

Hence as a matter of convenience, testing was done only to compare Proposal 1 and Proposal 2.

METHODOLOGY
As a preliminary testing process, a Pseudo Testing scheme was designed. A sample text of 25 pages was taken from an existing book and it was re-encoded according to Proposal 1 and Proposal 2 as show below.

Encoding:
Proposal 1 : Each Uyir and Mei character was encoded as a series of two bytes (8x2). The first byte would contain the Uyir or Mei character and the second byte would be blank.

e.g. அ = அ_ and க் = க்_

Each Uyir-Mei character was encoded as a series of four bytes (8x4). The first pair of bytes (16 bits) contains the Mei character and the second pair (16 bits) contains the Uyir character.

e.g. கி = க்_இ_ and = ச்_ஒ_

Proposal 2 : Each Uyir and Mei character was encoded exactly as in Proposal 1.

e.g. அ = அ_ and க் = க்_

However each Uyir-Mei character was encoded only as a series of two byte (8x2) characters.

e.g. கி = க்இ and சொ = ச்ஒ

The above two pseudo encoding schemes simulate the real situation fairly well.

The text derived from the above re-encoding process was used for testing various parameters that would affect the efficiency of the two schemes. For this purpose the following tests were carried out:

1. File size
2. Compressed file size using Pkzip
3. File copy using windows copy command (100 times)
4. Database Sorting of words from the text (20 times)
5. Database Indexing of words from the text (20 times)
6. Full word search for 'அவர்' in the complete text
7. Search for characters 'அன்' in any word in the complete text. e.g. in அவன்

TEST RESULTS
The results obtained from the above tests are tabulated below :

Sl.	Test	Proposal 2	Proposal 1	Difference
1.	File Size	116394 bytes	173904 bytes	49.41 %
2.	Compressed	35917 bytes	39467 bytes	9.88 %
3.	File Copy	1540 msecs	2080 msecs	35.06 %
4.	Database Sort	2310 msecs	3020 msecs	30.74 %
5.	Database Indexing	5490 msecs	7910 msecs	44.08 %
6.	Full word search	38450 msecs	58220 msecs	51.42 %
7.	'அன்' search	38010 msecs	57900 msecs	52.32 %

The pseudo test results are very clearly in favour of Proposal 2.

Other Languages: The concept of encoding all characters even if they are syllables, has been utilised by many languages. Primary examples are the Japanese Hiragana and Katakana script and the Korean Hangul Syllable block.

The Hiragana and Katakana Script allocates separate locations for syllable characters. e.g. 'ka', 'ki', 'ku', 'ke', 'ko', 'sa', 'si', 'su', 'se', 'so', and 'ta', 'ti', 'tu', 'te', 'to'.

Similarly, the Hangul Syllable block allocates a different location for each one of its syllables that are either a consonant-vowel-consonant combination or a consonant-vowel combination. In fact there are over 11172 such syllables which are allotted individual locations in Unicode (U+AC00 - U+D7A3). Apart from this the Hangul script also has a separate block called Hangul Jamo Blocl (U+1100 - U+11FF) which encodes the consonants and vowels alone without its combinations.

Another point to be noted is that the Canadian Syllabics have been alloted over 700 locations in Unicode 3.0

CONCLUSION:

Preliminary pseudo test results point clearly towards the All Character Encoding Scheme. But before proceeding further, it is necessary to test it out in the actual Unicode environment. This would require development of fonts and keyboard drivers. For this purpose the Tamils could come to a private understanding and use the End User subarea of the Private Use Area of Unicode (U+E000 - U+F8FF) for encoding all the Tamil characters. Once this testing is done, we would be in a position to take a final decision about how to proceed further. In case the results favour an All Character Encoding scheme, we should press further and get this implemented through the Unicode Consortium.

Author : The author is a partner of M/s Palaniappa Bros., which is one of the leading Tamil book publishing houses in Tamil Nadu. He is a Production Engineer with a Masters degree in Business Administration specialising in Finance and Information Systems. He has been involved in the fields of Font and Software development and DTP for over 15 years.

Contact : Palaniappan Bros.
14, Peters Road, Chennai - 600014, India.
Phone : 91-44-8268035, Fax : 91-44-8284067, eMail : chellappan@vsnl.com