Why Unicode ?

M.R.K. Murthy Raju
Nida Comm, Anna Nagar, Madurai-625 020, India <raju@murasu.com>

Unicode has been developed with a view to counter the problems and limitations inherent in the 8-bit representation of scripts.

With pre-unicode (8-bit) systems, language choice was linked to the default character set and locale chosen while installing the OS. Additional languages could only be brought in using additional software and having multiple languages inside the same document was not exactly easy.

Since Unicode uniquely identifies all the characters in all the major languages of the world, it becomes technically easier to represent text in multiple languages in the same document irrespective of the locale or the default character set of the OS.

Different methodologies

Based on the nature of individual languages and the way in which the native speakers of those languages looked at their scripts, different methodologies have been adopted while deciding on the code charts.

Alphabetical scripts like most of the Indic languages have been handled by thinking of the characters as combination of vowels, characters, vowel signs and some additional signs.

Non alphabetical scripts like Chinese etc. have been handled mainly based on the glyphs rather than on the underlying phonetic components of the characters

OS level support for Unicode:

Basic OS level support for Unicode for various languages including Tamil and other Indic languages is now available and is improving constantly with the open type initiative of Microsoft. For Linux and Other Unix like operating systems, also many solutions like PANGO, which are based on Unicode are slowly becoming available. This will certainly relieve the developers of the trouble of low level implementation details and they can concentrate on higher level applications like spell checkers, natural language processing etc.

Unicode and Tamil:

The Unicode scheme for Tamil has been based on the earlier ISCII standard evolved by C-DAC. This scheme has more or less uniform treatment for all indic scripts. This essentially treats the vowels,

consonants and vowel signs as the basic components of the script. As the Tamil has the benefit of a small character set under this scheme, the work of Tamil software developers, becomes easy to that extent. With all platforms supporting Unicode, this becomes the obvious choice for various applications like database storage etc. With the full charset falling within the 127 block, UTF-8 representation also becomes feasible.

Tamil and other Indic Scripts:

The possible benefits of sharing the same code structure with the other Indic scripts is obvious when we look at some of the software with support for all Indian languages like iLEAP from C-DAC, Pune. This software implements transliteration etc. with ISCII as the intermediate encoding. This makes transliteration a very easy task. Unicode can give that benefit to all Tamil software.

The Unicode methodology used for Indic scripts is very intuitive and conforms to how we represent our script internally. Hence this can be very helpful while implementing language based solutions.

The present arrangement has other great benefits like easier sorting and faster implementation etc. Asking for a different unicode representation for Tamil, where every possible combination has its own code value, will negate these benefits.

There are also other benefits if we consider all the bilingual Tamils who know or work on at least one non Tamil indic language/script apart from Tamil.

There may be other non technical, but more convincing economic benefits for Tamil developer community in taking other Indian languages along. Since , All Indian scripts, except probably for Urdu, have the same structure , same code values differing only in the bit representing the language. porting the current Tamil software products to other Indian languages/scripts leveraging on Unicode will be very very easy. With their early lead and rich experience, Tamil developers are in the right position to play a lead role in helping other Indian languages also on the net. This will not be possible or so easily possible if Tamil has its own unicode structure different from that of the other Indic scripts.

Arguments against Unicode:

Since Unicode is essentially a 16-bit representation, unicode text tends to occupy more space compared to the text represented in 8-bit encoding systems. There are many ways in which we can counter this problem, one of them being UTF-8. And with cheaper disk space and faster processors, the size-based argument is slowly becoming less convincing.

Considering all the benefits, we need to give a serious thought to the usefulness of Unicode in general and the present structure for Tamil in particular.