L2/00-100

ISO/IEC JTC/1 SC/2 WG/2 **N2198**

2000-03-15

ISO/IEC JTC/1 SC/2 WG/2
Universal Multiple-Octet Coded Character Set (UCS)
Secretariat: ANSI

Title:	**Proposal: TR 15285 ammendments**
Doc. Type:	National body contribution
Source:	Japan -tks
Project:	02.18
Status:	**For consideration at Beijing WG2 meeting**
Date:	2000-03-15
Distribution:	SC2 WG2
Reference:	
Medium:	

This document proposes an amendment for the ISO/IEC TR 15285 “An operational model for characters and glyphs”. The amendment may have two annexes for the TR.
One discusses about a “character repertoire design guidance for Syllabic Scripts” and another discusses about a “recommended functionality for supporting sophisticated input assistance”. Both annexes are independent to each other but have strong interrelation between the two.

Problem description:

1. Most of the syllabic scripts needs to have a rendering process of a string of multiple coded elements for presentation of them. And also, sometimes, a string unit of the coded elements should be handled as if it is one single character for data processing purpose, and sometimes it requires a single normalized sequence of the coded elements (for processing, such as sorting) even though human being would type-in the sequence differently.
   Therefore, there is a needs of well-defined “processing unit of coded elements” to handle the most of the syllabic scripts for machine process and presentation purposes.
   In addition to that, even though the “unit” is defined, glyph shape is not always defined as one shape. Sometimes, there are multiple candidate shapes are available for the one single unit. This is another cause of ambiguity.

On the other hand, the selection of the repertoire (or coded elements) is normally done by linguistic viewpoint and under strong influence of handwriting culture of the script. This is quite natural.

Because, each elements for handwriting are defined assuming super interagency (human being) who is able to do what simple machine (such as computer) cannot do. To discriminate the “unit (sequence) of coded elements for processing and or presentation” from long stream of coded elements by human being might be easy task, however, sometimes it is difficult to do so by machine in automatic discrimination algorithm. (For some scripts, it is easy)
For detail explanation of this problem, see WG2 N2148 attachment by Dr. Tashiro of ETL-Japan

For understanding of the problem, assume that there are the coded element <A>, <B>, <C>, and also, there are the word <A>, <B>, <C>, <A><B>, <A><B><C>, <A><B><C><A> and <C><A>. And assume the presentation shapes of each word are different (not like combination of <A>and <B> and so on. If string ABCABCABCABC is received, how the presentation of the shapes were select the shapes for presentation? It may be <A>, <B>, <C>,……., also, might be <A><B>, <C><A>…… or it may be <A><B><C><A>…… It is also difficult to search <A><B>, because <A><B> is not a <A> and <B>. How about wrap around for end of line? Any of super sophisticated rendering (and search) machine can not utilize it’s super power unless someone out side the machine defines the units for render or process.
And <A><B> can be connected side by side or stacked vertically. Either might do a job, but sometimes, the difference in shape means different meaning and/or pronunciation.

Therefore, special consideration for repertoire (or coded elements) selection is necessary to satisfy the requirements of machine presentation/process of those scripts. It is necessary to have computer process consideration on top of the linguistic solution.

One of proposed annex should describe this problem and also should recommend several choices of the solutions for the problem.

2. Once the code is designed for machine process convenience to avoid the problem described above, it is not necessary that the resultant coded element is human friendly. It would be also true that since the code is changed for machine convenience from natural writing method, the possibility would be very high that the resultant code is human

un-friendly from input (or operator) viewpoint. (Due to the reason why it is developed, the output should be human friendly already). If a string of the coded elements should generated by the human being directly (like one by one key-in), the input would not be accepted by the user because it is not human friendly. This is why some of the code table made compromise with the human friendly input and machine un-friendly process. And it ends up at special process requirements specific for the script. This is not a wise goal of the internationalization (i18n).

Another annex is providing a solution for this problem. If there were good “input assistance” (instead of direct key-in operation) such as IME, the problem would be resolved easily.

Code within machine may not be human friendly (machine friendly), but the generation method can be human friendly. Therefore, user of the system can not feel any unconvinced because of human unfriendly internal coded elements.

Proposed Annexes.

The scope (or general view) of the annexes are as follow:

Annex-X: This annex is dealing with the problem-1 above. It will be describing:

1. Problem explanation with sample (problem due to the ambiguity)
2. The reason of the problem.
3. Rule for coded element selection: It should say that the coded elements should be selected such that any ambiguity to discriminate the “unit for render/process”. The linguistic/cultural correctness and human friendliness should have less priority than elimination of the ambiguity.
4. Hints for code design: If needed, consider special approach such as “Introduce special discriminator/joiner/terminator”, “make them as pre-composed”, “assign multiple code points for a single shape (like independent and composite and so on…)” or “use sub-composite sequence approach for some elements” etc…..with sample explanation(s).
5. Describe some of related requirements as described in the paper by Dr. Tashiro.

Annex-Y: This annex is dealing with the ploblem-2 above. It will be discussing:

1. Problem explanation with sample (problem due to the Annex-X approach)

2. Solution for the problem (the new input assistance will resolve the problem)
3. Sample (or hint) of the solution (Japanese input method)
4. Requirements for the new approach

We understand that:

There are many state of the art rendering technologies are almost ready to use, however, what this annexes is going to resolve is an issue about the data themselves which to be used by the rendering engines. If there is an ambiguity of the data, whatever the engine can do, it is not possible to do right rendering. No question about the proposal should be written considering the most sophisticated state of the art rendering engine

This does not fit the ISO/IEC TR 15285 (current) scope directly, however, since it is tightly related and resultant deliverable will be a coded character set(s), therefore, to combine then as a single document would make sense.

Note: this is a result of joint activity of more than 15 Asian countries under MLIT-program.