L2/01-160
ISO/IEC JTC 1/SC 2/WG 2 N2359
05 April, 2001

ISO
International Organization for Standardization
Organisation Internationale de Normalisation

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

Title:	Ad Hoc Discussion on Amending the Character-Glyph Model (TR 15285: 1998)
Source:	Edwin Hart, Takayuki K. Sato, Ken Whistler, Christian Cooke
Status:	Expert contribution
Action:	For information to SC 2/WG 2
Distribution:	ISO/IEC JTC 1/SC 2/WG 2
References:	WG 2 N2148, N2198, N2199, N2206, N2319 (Appendix A), N2324

At WG 2 Meeting 40 in Mountain View, USA, the authors met to discuss requirements for amending TR 15285 to accommodate the requirements in the referenced documents. These notes summarize the plan to amend the TR.

Working Paper for Proposed Modifications to TR 15285: 1998

Scope of Amendment

The goal is to revise ISO/IEC TR 15285: 1998 to discuss (1) character input-methods and (2) principles and issues to be considered in deciding how to code a particular script.

Justification

The goal is to enhance TR 15285 so that it will provide guidance for coding scripts that have not been coded before and do not have an established coding model/paradigm. The principle concern is for uncoded scripts used in Southern and Southeast Asia. The goal is to document a set of issues and considerations so that people avoid developing new coded-character-sets that are incompatible with ISO/IEC 10646 principles so that the new codes cannot be easily added to ISO/IEC 10646. The document should discourage inventing unnecessary new coding models. People developing new codes need to understand that the coding model does not directly depend on how users will enter the text and how characters need to be rendered for display and printing. While the coding model may be closely related to text entry and rendering, each has different requirements and therefore each set of requirements needs to be considered separately.

Coordination with SC 35

Because of the discussion on keyboard input methods, SC 2 should coordinate the effort with SC 35.

Strategy

Before requesting formal WG 2 and SC 2 approval to create the amendment, the editor plans to draft a rather complete set of text changes to the TR to ensure that he understands the issues and scope of the effort. Implementing this strategy will increase the editor's confidence that this effort can be successfully completed. The editor will use the several papers submitted by Japan to understand the requirements and the topics to include in the amendment.

Specific Changes

The following are the initial ideas of the expected changes. However, the editor requests the flexibility to make changes, where appropriate, to cover the new scope.

EDITORIAL CHANGES

1. Reduce the dependence of the document on the glyphs in the specially created fonts from Michael Everson and Kamal Monsour, and in other fonts with limited availability. To the extent possible, replace these glyphs with glyphs from commercial, Unicode/10646 fonts.

TECHNICAL CHANGES

1. Add text to section 5.2, Composition, layout, and presentation, to describe the data entry process. Data entry is one of the boxes in Figure 2, which is described in this section. Simply expand the section to describe the data entry process, and in particular input methods. Voice input is out of scope. Include the following important points:

    a. While computer input on keyboards appears to be the same as typing on a typewriter, translating keystokes into computer codes can get quite complex. Unlike typing on a typewriter, the relationship between the character displayed on the keytop and the character input into the computer or displayed on the screen can be quite complex.

    b. An input method maps keystrokes into character codes.

    c. The input method converts typestrokes into characters by converting a series of keyboard codes of the struck keys into a series of coded-characters.

    d. Like rendering characters into glyphs, the mapping from keystokes to characters may be one-to-one or quite complex.

    e. Characters can be typed in either logical order or display order in cases where the display order is different from the logical or pronunciation order. With some scripts, some users may have been taught to write characters in one order and others in a different order, depending on when and where one convention was taught over the other one. In such circumstances when users of a script use both typing conventions, the input method must allow users to select and use the input convention they prefer. It may be desirable for the input method to automatically recognize the convention rather than forcing the user to select the input convention.

    f. ISO/IEC 10646 establishes the convention of whether a script orders coded characters in either logical or displayed order (but not both).

    g. Regardless of whether the keystrokes are in logical or display order and regardless of whether the coded characters in a script are stored in logical or displayed order, the input method converts the keyboard codes into properly ordered coded-characters for a given script.

2. In the same way that TR 15285 includes separate annexes to expand the discussion on characters and glyphs, add a new annex to provide more details and examples of input methods. This annex should likely follow the annexes for characters and glyphs. Include the following important points and examples:

   a. Discuss Issues with Input Methods

      1) To satisfy the user requirements, input methods depend on the script, the language and the culture. Input methods generally follow four distinct models:

         a) Entering characters from a repertoire much much larger than the number of keys on a keyboard. This is the issue with the repertoires of East Asian ideographic scripts. Input methods include phonetic and stroke or radical input. To remove ambiguity, these methods generally require using a dictionary to present the user with alternatives and having the user select the desired ideograph.

         b) Entering characters from scripts with complex rendering rules where the input method may need to be closely tied to rendering. This applies to the complex scripts of Southern and Southeast Asia. With these scripts, a key may not equal a glyph, and typing the next key may require that the glyph be changed. While the input and coding for such scripts follows a logical order, the set of glyphs, the mapping from characters into glyphs, and glyph placement are complex.

         c) Entering characters for the bi-directional (bi-di) scripts with complex rules for reordering codes for displaying and printing. Codes for these scripts are stored in logical order but the display order varies. This is not a coding issue nor a data input issue. While the coding and rendering characters into glyphs is straightforward, deciding the display order depends on complex rules and must be done as the user types new characters. Example scripts include Arabic and Hebrew.

         d) Entering characters where the size of the character repertoire is similar to the number of keys and input is straightforward because the issues in the three previous models do not apply. Examples include keyboards for the Latin, Greek, Cyrillic, and Korean Hangul scripts.

            Entering characters from a different repertoire (script) than the one shown on the keytops, e.g., entering Greek characters from a US English keyboard for the Latin script.

      2) Many different commercial input methods exist. Input methods should not necessarily be standardized. Users need choices and stability. Moreover, vendors need the freedom to compete by developing enhanced input methods to differentiat one product from another.

      3) Input methods should not require users to do unnatural things merely to type text into the computer (verus using a typewriter or handwriting).

   b. Examples:

      1) Typing SMALL E WITH ACUTE to generate the 10646 composed character or combining sequence (reversing the typing order of the combining accent and the base character):

         a) With a keyboard with one key with “é” on the keytop.

         b) With a keyboard with a dead key for the acute accent and a separate “e” key.

      2) Typing Japanese Kanji ideographic characters using a keyboard with phonetic Romanji keys.

      3) Typing Chinese Hanzi ideographic characters using a keyboard with radical keys.

3. Add new sections to Annex B, Characters to describe to describe principles for deciding what to encode as characters. Annexes B.4 and B.5 have started discussing such principles but are insufficient. Include the following important points:

   a. The goal is to create codes according models that are as simple as possible yet still satisfy the requirements.

   b. Sorting is an issue that is separate from coding (the assignment of code points (values) to characters). Add references ISO/IEC 14651, and UTS 10, *Unicode Collation Algorithm.*

   c. Store characters in logical order versus display order.

   d. Example from Bengali where the glyph for some combining vowels appears on both sides (left and right) of the base character. The idea is to code one character rather than two.

   e. Repeated fragments of characters: English “n” and “m”, and “v” and “w”. Hangul Jamo has syllables with similar repeated fragments (show “T” and “?” like fragments in Sato-San’s example). If three of these in sequence, e.g., “vvv”, how do you distinguish between “v-w” versus “w-v” sequences without a dictionary?

   f. Consider typewriter or coded-character-set if either or both exist for guidance but do not rely on them to the exclusion of other knowledge of the script.

   g. The convention for storing characters in a script should not depend on the input order, and shall not depend on the input order for scripts where the keying order includes both the logical and display order conventions.

   h. Describe coding models for Devanagari, Thai/Lao, and Tibetan.

      All three models make extensive use of combining marks. The ISCII/Devanagari model is used to encode all other Indic scripts, as well as Sinhala, Khmer, and Myanmar. (And is the preferred model for newly encoded Brahmi-derived scripts, unless there is a compelling reason to do otherwise.) The Tibetan model is used to encode Tibetan. The Thai model is used to encode Thai and Lao.

      1) The ISCII/Devanagari model

         This uses virama to encode consonant conjuncts. It uses logical order for all characters, and encodes no duplicated characters for "half" character forms, conjunct parts, or special forms of RA, WA, YA, LA, HA, etc. It encodes a separate series of independent vowel letters and a separate series of dependent ("matra") vowels.

      2) The Tibetan model

         This does *not* use a virama. It uses logical order for all characters, but encodes a separate series of "subjoined" consonants to deal with consonant combinations. It has only a single series of vowels, which are all dependent.

      3) The Thai model

         This uses display order, left-to-right, rather than logical order, since it was developed based on typewriter technology. In practice, this means that a small number of "left-side" vowels must be rearranged by processes such as collation, to get correct results based on the logical order of syllable sequences.