4S－3

Multilingual I/O and Text Manipulation System (1): The Total Design of the Generalized System based on the World's Writing Scripts and Code Sets

Yutaka Kataoka*, Tadao Tanaka†, Kazutomo Uezono†, Tomoko Kataoka* and Hiroyoshi Ohara†
* Center for Informatics, Waseda University † School of Science and Engineering, Waseda University

1. Introduction

International networkings with a lot of international/national code sets spread as common environment of daily use. And users of different nationalities have shared one computer on such environment. Thus, it is essential to process and to communicate texts in multiple code sets simultaneously and consistently.

On the other hand, current approaches to *Internationalization* (I18N) are mainly adapting a system to one specific language based on *POSIX Locale Model* [1,2], which is only a localization to a bi-lingual system. Since POSIX does not inhibit to define multi-code-set (MCS) Locale, it is possible to set MCS Locale. But the Locale reveals contradictions of the Locale model[3].

To be an I18N system, four components, I/O, Text Manipulation/Communication (TM/C) are essential, and all components must ensure computability and interoperability in unlimited use of set of MCSs. Then all the functions work consistently by defining each element of MCSs after extensions (in *Perso-Arabic scripts*, *Final Glyph Set* to be displayed is larger than its *Graphic Character Set*[4]).

As a trial in a part of I18N, X11R5 OM (Output Method) and IM (Input Method) were designed and implemented[5]. In R5, a set of limited MCS was provided. But in R6, its OM and IM are hard-coded to a locale. R5 and R6 do not ensure computability, and are not I18N system by the limitation from the Locale model. Also ICCCM does not ensure interoperability that occurs from code set design.

Thus, new I18N model that satisfies MCS Locales (*Multi-local Model*) was established by our researches based on world's writing scripts and code sets. And for satisfying TM/C and essential factors out of ISO specifications – beyond the new I18N model – more general model (*Global IOTMC Model*) covering the four components was established and the Multilingual I/O and TM/C System was newly developed.

2. Extent and Definition of Multilingual I/O and Text Manipulation/Communication System

A multilingual I/O and TM/C system should have four components, 1) Input, 2) Output, 3) Text manipulation and 4) Inter-process communication. And in each component, following conditions must be satisfied, 1) unlimited multi-code-set must be handled simultaneously, 2) all code sets (including control character sets) must be specifiable by ISO 2022[6], 3) extended codepoint by ISO 6429[7] must be specifiable, 4) beyond ISO specifications, final glyph set of non-fixed length codepoints (ex. TIS 620-2533:1990 – *Thai* script, IS 13194:1991 – Scripts in India) must be specifiable, 5) conversion between mb and WC must be ensured including the case of non-fixed length codepoints (NFLC), 6) text manipulation functions must be code set independent, 7) informations for inter-process communication must be supplied for an application. And to ensure computability, each element in each code set must be specifiable after trans-code-set conversion with codepoint extensions. Definition of multilingual I/O and TM/C system can be defined as a system that satisfies the above requirements. Thus, a system that cannot define extension rules beyond ISO to determine each element in each set is not multilingual I/O and TM/C system.

3. Definition of Generalized Multilingual I/O and Text Manipulation/Communication System

To process texts in MCSs unlimitedly, it is essential to have data files that describe definitions of all code sets and extension rules without any hard-coded part in a system. To do so, only one structure executing each component of the four is required. Then WC should be one set through all locales, and so be final glyph set. Thus, the definition is satisfying above three requirements.

4. Classifications of World's Writing Scripts and Code Set Designs

To design generalized multilingual I/O and TM/C system, world's writing scripts were analyzed and classified[5]. Writing scripts in the world can be basically classified into the following categories, 1) *Phonemic*, 2-1) *Conjunct Syllabic*, 2-2) *Pure Syllabic* and 3) *Ideogrammic*. In order to specify a final glyph, writing scripts are also classified into following, 1) *Direction dependent* and 2) *Position dependent*. And writing scripts can be classified into 1) *Fixed writing direction* (ex., *Mongolian*) and 2) *Non-fixed direction*. To specify a final glyph shape, informations that specify *Writing Origin*, *Script Direction* and glyph shapes in positions in a word are required. Thus, final glyph set must be determined as computable by explicit rules that are not described in ISO specifications.

To determine codepoint extension methods, classification of national/international code set designs is also essential. Code set designs are classified into the following categories, 1) 1 codepoint for 1 glyph, 2) 1 codepoint for multiple glyphs, 3) multiple codepoints for 1 glyph and 4) multiple codepoints for multiple glyphs.

By the classifications of writing scripts and code set designs, it is clear that 1) one codepoint of mb or WC does not stand for one character, 2) minimum unit of mb/WC sequence exists to process correctly, 3) explicit rules to convert among mb/WC/final glyph sets must be defined to be computable (WC is an implementation dependent code).

5. ISO Code Set Extensions and Essential Extensions beyond ISO

By the analyses above, at least three code sets, mb, WC and final glyph set must be provided. In the conversions among them, ISO 2022 specifies only encoding schemes and ISO 6429 specifies extra set of characters. Adding to ISO specifications, non-one-by-one mapping of codepoints and glyphs, Position dependent and Direction dependent require other extensions beyond ISO. Thus, adding to encoding scheme conversion, *Trans-Character-Set Conversion* is required.

IS 13194:1991 requires multiple different rule sets of extensions according to writing scripts derived from *Brahmi Script*. This kind of extension rule set change is out of ISO 6429. Thus, inter-process communication designed by ISO 2022 and 6429 does not work. Thus, mechanism that specifies combination between graphic character set and rule set is required.

6. The WC, Its Redefinition and Limitation

WC is an implementation dependent code converted from mb by POSIX. When one WC codepoint is always converted from one mb codepoint, POSIX wctypes functions do not work

correctly. By the analyses described above, complete conversion between mb and WC including multiple mb codepoints to WC is required for the extensions. To use WC generally, WC must be a user definable code converted from mb by explicit rules and WC codepoints must be unique through all locales to avoid locale dependency. And to clearly define WC, a codepoint of WC having a glyph must be mapped to one codepoint of the final glyph set. Thus, WC contains codepoints one-by-one-mapped to the final glyph set and non-displayed codepoints introduced from ISO control character sets and from Non-ISO control character set.

But still WC does not ensure to stand for one character. And there are multiple ways to normalize to one character, thus, for text manipulation, WC cannot be used.

7. Introducing TMC and Generalized Text Manipulation Functions

As described above, WC is not a set of characters. To generalize text manipulation in multilingual text, a code that is normalized by character to be processed by absorbing code set design differences – *Text Manipulation Code* (TMC) is required. To avoid reducing computability, a mechanism that provides multiple TMCs from WC by different normalizations must be provided. By our researches, basic unit of character to be processed was discovered, ie., it is possible to determine one character of all of scripts in encodings/extensions. And basic functions to manipulate text strings and techniques to generalize the functions to TMCs were discovered.

TMC one codepoint has *Character-ID field* and *Attribute field*. The Attribute field retains bits of categories of a character, eg., punctuation symbol, phonemic, position dependency and so on. Those informations are definable and described in WC-TMC conversion tables. Especially, TMC-ID 0 is provided by our system normalized by character for the purpose of basic multilingual text manipulation. By using TMC-ID 0, widgets are now rewritten.

By the informations in the Attribute field, text manipulation functions could be designed as TMC independent, ie., the text manipulation functions use both or one field for the purposes.

8. Multilingual Inter-Process Communication

In inter-process communication, ISO specifications do not supply informations for locale. And locale dependent factors are all implementation dependent. Thus, a locale dependent system is closed. Therefore, it is clear that a model independent from Locale model is required. But such a process is inadequate that has all the encodings and the extensions. Those informations should be returned from the multilingual system.

Not only returning the informations but also associating a code set to a rule set to generate WC and other code sets are essential – IS 13194 can be used for all different scripts derived from Brahmi.

9. The Multi-Locale Model and The Global IOTMC Model

To satisfy conditions and analyses described above, two models were established. The Multi-Locale Model provides MCS locales by *OpenLocale* function that returns Locale-ID. By using of the ID, the system get I/O MCS simultaneously. Function *setlocale* and locale-related I/O functions are kept for backward compatibility. In this model, only one WC for all locales is provided. Thus, it is possible to share WC codepoints among different locales.

On the other hand, The Global IOTMC Model covers all of I/O and TM/C. This model provides TMC-ID 0, basic text manipulation functions and inter-process communication functions. By this model, it is not necessary to open locales. Once calling *InitGlobal* function to be initialized, all graphic character sets and control character sets can be used without locale-ID. And this model also provides switching mechanism that associates a graphic character set and its extension rules (this mechanism solves IS 13194 and Perso-Arabic Scripts problems). By the optimal implementation architecture of the models, all functions of the models can work simultaneously.

Note that specifications in X11R5/6 for text drawing and for text extents do not satisfy vertical and bi-directional drawings. Thus, we redefined the specifications more generally and add extra functions to satisfy the vertical and bi-directional drawings.

10. The Architecture of the Models

The architecture of the models is based on *Meta Converter System* that converts encoding schemes, Trans-character-set and *Trans-Unit*, since all sets must be defined clearly to keep computability. The system generates WC and IWC (Internal Wide Character specifying final glyph set) from mb with rules described in the tables. And the system generates TMCs from WC. The system also converts TMCs to WC and WC to mb when conditions for reverse conversion are satisfied.

The Meta Converter System is a complex of automata and each automaton is generated from *Meta Converter Table Compiler*. By the analysis of writing scripts and code sets, optimal paths and structures of the automaton complex and each automaton were discovered. To minimize code of the system and to make it fastest, basic extension functions were discovered and implemented. Thus, the automata call the functions with rules. Therefore, the best performance is ensured. By memory sharing functions, all common area of the system is shared by processes.

Each functional part is designed as a module to call the Meta Converter System. Thus, each element of a set is always ensured as unique. By this architecture, total system can keep computability and interoperability.

11. Summary

A Multilingual I/O and TM/C system was developed by Multi-Locale Model and Global IOTMC Model. The architecture of the system is based on Meta Converter System. By the analyses to develop the system, problems of POSIX and ISO specifications were solved. Also by our researches, requirements of mapping a set to others became clear.

References

[1] ANSI/IEEE Std 1003.1-1998, IEEE Standard Portable Operating System Interface for Computer Environments (Approved Nov-10,'89 by ANSI).

[2] ISO/IEC 9945-1: 1990, Information technology – Portable Operating System Interface (POSIX) Part 1: System Application Program Interface (API) [C Language].

[3] Kataoka, Y. et al., A model for Input and Output of Multilingual text in a windowing environment, ACM UIST'91 November 11-13, pp 175-183.

[4] Tanaka, T., et al., Generalized Output System that draws multiple code sets on a windowing environment (in Japanese), Proceedings of the 46th General Meeting of IPSJ, vol. 5, March 1993, pp 87-89.

[5] Kataoka, Y. et al., A model for Input and Output of Multilingual text in a windowing environment, ACM Transactions on Information Systems, Vol. 10, No. 4, October 1992, pp 438-451.

[6] ISO/IEC 2022: 1986, Information processing – 7-bit and 8-bit coded character sets – Code extension techniques.

[7] ISO/IEC 6429: 1988, Information processing – Control functions for 7-bit and 8-bit coded character sets.

[8] TIS 620-2533 (1990), Thai Character Codes for Computers, Thai Industrial Standards Institiute, Ministry of Industry, Thailand.

[9] IS 13194:1991, Indian Script Code for Infromation Interchange – ISCII, Bureau of Indian Standards, India.