The Multilingual Text Processing (5):

1Q－5

Extension of POSIX to Multilingual Processing

Kazutomo Uezono†, Tomoko Kataoka*, Toshio Oya†, Hidejiro Daikokuya†,
Dawa Yidemucao†, Yutaka Kataoka*, Hiroyoshi Ohara†

* Media Network Center, Waseda University † School of Science and Engineering, Waseda University

1. Introduction

Most systems have been constructed based on *POSIX Locale Model* [1, 2, 3] or *limited multilingual models*, which do not fully support ISO 2022 [4]. Especially, the system based on POSIX Locale Model does not work correctly at an interactive locale selection due to the ambiguous POSIX specifications which do not have farther interpretations [9].

To process simultaneously all scripts in the world, *Global IOTMC Model* was established by the analyses of *Character*, *Codeset* and *Text processing* in the world [7, 8], which supports ISO 2022 and other code extensions beyond ISO specifications. Since a system internal code (*WC*) was normalized to one Character and includes not only an identifier but also the drawing information, etc., WC is converted uniquely without locale dependencies. As a result, all of the codesets can be processed simultaneously. And, the relation between this model and POSIX Locale Model was defined clearly for keeping backward compatibilities.

The System 1, developed based on Global IOTMC Model by Waseda University, provides the internationalized computing environment.

2. POSIX Locale Model

POSIX Locale Model specifies that the Locale table includes information of English and another language. It supports multiple-language processing by switching the Locale (Fig. 1). Since the results gotten by the Locale related functions make sense in that Locale, the Locale cannot be switched on a process. Therefore, only one Locale is provided for an application, which shows that this model is no more than bilingual model even if it includes many Locales.

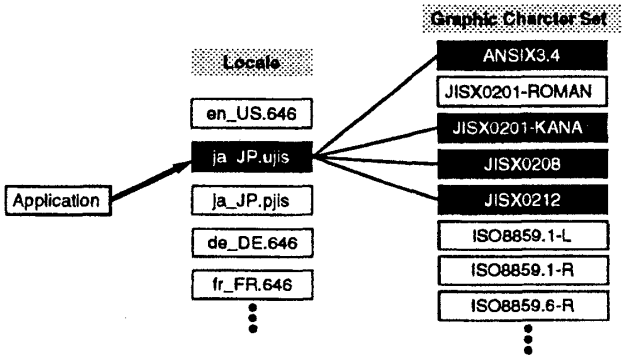

Figure 1. POSIX Locale Model

3. Extension to Multilingual Model

By the analyses of Character and Text processing, it became clear that Text processing is classified into 1) *Language-independent processing* and 2)*Language-dependent processing*. Insertion, Deletion, Search, Replacement and Line Feeding are typical of 1), which needs only Character information. On the other hand, Spell checking and Hyphenation are typical of 2), which needs Language information besides Character information.

Since the sets of Characters in 1) were defined clearly, it was possible to operate Language-independent Text processing correctly without Locale dependencies. Furthermore, Language information was added to each element in these sets, that enabled to do Language-dependent Text processing without Locale dependencies. As a result, all texts can be processed simultaneously and *the Multilingual text processing* can be realized.

4. Definition of WC and TMC

Mb is a set of *Graphic Character Sets* (GCSs) and *Control Character Sets* (CCSs). Since an mb codepoint does not always stand for a *Final Glyph* nor a *Control Function*, mb codepoints are *Non-Fixed-Length* (NFL) codepoints. The mb codepoints are extended to *Extended Codesets* by ISO 2022 and combination of codepoints in GCSs and CCSs for codepoint extensions (ISO 6429 [5], IS 13194 [6], etc.).

POSIX defines the Extended Codeset as WC. But mb/WC conversion is the type conversion from *char* to *wchar_t* in POSIX, which causes impossible to extend by ISO 6429, IS 13194, etc. Since the Extended Codesets are mapped to WC, a WC codepoint must satisfy the following requirements: 1) to have information for presentation, 2) to have information for basic text manipulation(Insertion, Deletion, etc.), 3) to keep codeset/language independency, and 4) to ensure reverse-conversion. To satisfy the above, WC is normalized to a Character. The set is the biggest and involves all information of GCSs and CCSs. WC is unique in a system to keep consistency among all models.

WC can be used for text manipulation without codeset/language dependency. To manipulate text codeset/language dependently or glyph dependently, *Text Manipulation Code*(TMC) is defined. TMC is converted from WC with necessary information and/or glyph information given by *Output Method* (OM). *Final Glyph Set*(FGS) are mapped from WC.

An example of mapping for Devanagari script is shown Figure 2, and the overall relations among mb, WC, TMC and FGS are shown in Figure 3.

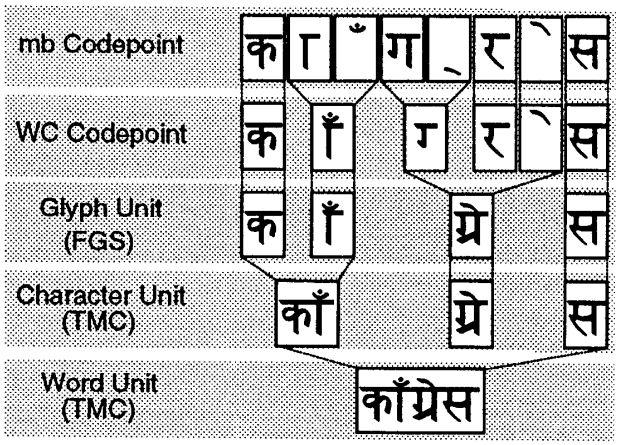

Figure 2. Process Unit in Devanagari script

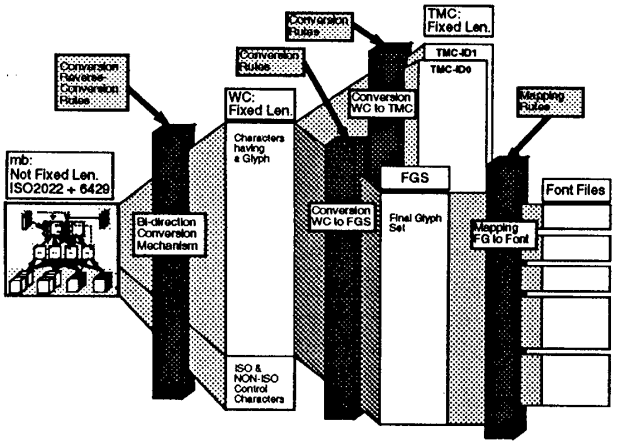

Figure 3. Relations among mb, WC, TMC and FGS

5. Establishment of Global IOTMC Model and Multi-locale Model

By redefinition of WC as above, two new models beyond POSIX Locale Model, named *Global IOTMC Model* and *Multi-locale Model*, were developed (Fig. 4).

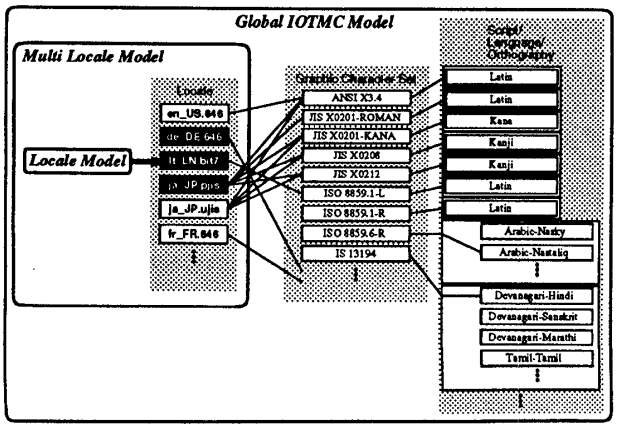

Figure 4. Relations among Global IOTMC Model, Multi-locale Model and Locale Model

Global IOTMC Model does not have a default state, which supports not only ISO 2022 but also code extensions of other International/National specifications and code extensions derived from analyses of Characters, Languages and Orthographies. POSIX Locale Model is only a subset of Global IOTMC Model, because it only specifies a default state. For *Interprocess communication*, Multi-locale Model specifies some Locale tables simultaneously.

6. Summary

The System 1, developed based on Global IOTMC Model, provides Multilingual Input/Output/Text manipulation/Interprocess communication. By replacing Locale-specific functions with the System 1's, *Operating system* or *X Window system* are enhanced to the real multilingual system (Fig. 5), which enable to develop multilingual utilities such as *Formatters*, *Parsers* and *Printing*. It will contribute greatly to the progress of the international computing.

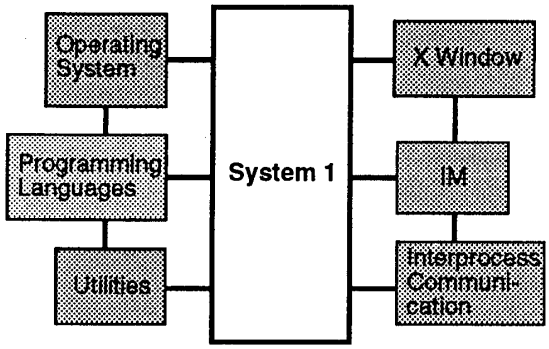

Figure 5. The System 1 Environment

References

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

[2] ISO/IEC 9899: 1990, Programming languages – C.

[3] ISO/IEC 9899: 1990/DAM 3, Draft Amendment 1: 1994 (E), Programming languages – C AMENDMENT 1: C Integrity.

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

[5] ISO/IEC 6429: 1992, Information technology – Control functions for coded character sets.

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

[7] Kataoka, Y., et al., 1995. Codeset Independent Full Multilingual Operating System: Principles, Models and Optimal Architecture, IPSJ SIG System Software & Operating System, 68-4, pp 25-32.

[8] Uezono, K., et al., The Worldwide Multilingual Computing (2): Functions, Model, Design and Architecture of Multilingual I/O TM/C System, Proceedings of the 51th General Meeting of IPSJ, Vol. 3, September 1995, pp 247-248.

[9] Yamanishi, S., et al., The Worldwide Multilingual Computing (8): Multilingual Basic Environment – C Language and OS, Proceedings of the 51th General Meeting of IPSJ, Vol. 3, September 1995, pp 259-260.