4Q－1

The Worldwide Multilingual Computing (1): Essentials, Principles and Scope Covering All Characters in the World

Yutaka Kataoka*, Kazutomo Uezono†, Tomoko Kataoka*, Tadao Tanaka‡, Toshio Oya†, Hidejiro Daikokuya†, Kenji Maruyama‡, Shoichiro Yamanishi† and Hiroyoshi Ohara†

* Centre for Informatics, Waseda University † School of Science and Engineering, Waseda University
‡ Research and Development, Japan Computer Corporation

1. Introduction

For all the computings placed in strong requirements of multilingual processing, a total multilingual computing environment has been waited for a long time – POSIX Locale model and limited multilingual model have brought serious obstacles [1, 2]. By analyses and generalization of all characters in the world, the first true Multilingual Computing Environment (WASEDA MLCE) [3] was designed and developed that supports any processing of texts in simultaneously mixed all character codesets and all languages (Fig. 1).

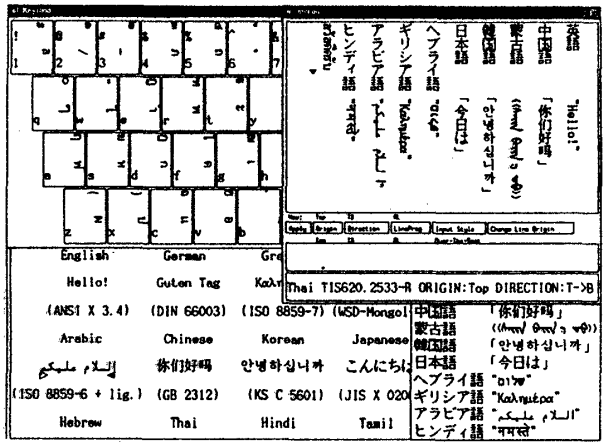

Figure 1. A Display of WASEDA MLCE

The environment was provided by *Global IOTMC System* that realizes overall multilingual environment including I/O, interprocess communication, text manipulation and programming languages on a system without any inconsistency. By the discovery of the definition of *Character* [3], the kernel of the system named *Meta-Converter System* [4] not only absorbs dependencies of characters, codesets, languages and others but also realizes generalized text processings with specific information of such dependencies for advanced processing [5].

The interfaces of the system were designed based on Global IOTMC model that ensures backward compatibilities to historic localized models [6, 7, 8] with mixing all national/international codesets [9, 10]. The environment can provide an application that does not contain hard-coded part for dependencies above. Thus, the system dramatically contributes language educations, databases, text processing, international networkings and so on. Adding to those, the environment provides multilingual natural language processing by generalized text manipulation functions [3, 5].

2. Essentials: Definition of Character, WC and TMC

In general, *Glyph* and *Character* are considered as the same. But a shape of a character is changed according to its position in a word and writing direction. Thus, character specifies a set of glyphs – logically 16 final glyphs (4 directions × 4 positions). By this, a character can be defined as a name of a set of glyphs [3](Fig. 2).

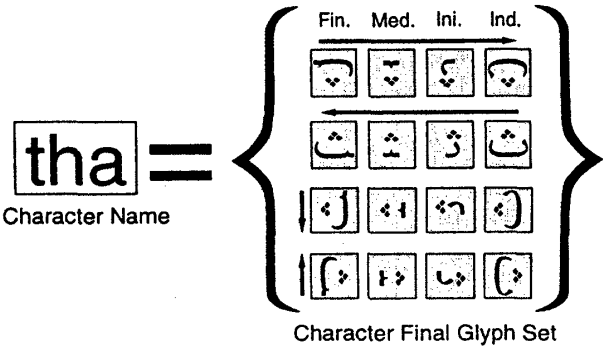

Figure 2. A Character is a name of a glyph set

Characters in the world are categorized into the following: 1) *Phonemic*, 2) *Syllabic* and 3) *Ideogrammic*. Syllabic contains 2-1) *Pure*, 2-2) *Conjunct* and 2-3) *Phonemic-Syllabic* [3]. In any case, determining one character was discovered possible. Note that multiple ways exist to determine one character for 2-2.

Characters to be processed on a computer system are listed in character codesets. But one *Codepoint* does not stand for one *Character* by *Extension methods*. All Extension methods were expressed by limited number of *Conversion functions* [3, 11]. Thus, character codesets should have rules to extend and can be classified into 'one character specifiable' or 'not'.

When all codesets are 'one character specifiable' it is possible to convert into WC, an internal codeset, i.e., one codepoint of WC can be a Character. This definition of WC brought generalized text manipulation including line separation that does not depend on codesets, glyphs and languages. Note that a function of OM is determining final glyphs from sequence of WC by rules.

To process a character/unit with dependencies of character, codeset or language, TMC [5] was introduced. TMCs are converted from WC with rules for personal requirements. By using named bitfield in TMC, it is possible to manipulate different TMCs by the same set of

manipulation functions [5].

3. Principles: Requirements and Conditions

Before multilingual system came, transliteration brought a lot of troubles, e.g. *Thai* has many characters having the same phonetic values. Thus, multilingual computing must simultaneously process all character codesets of both national and international standards except for miss designed codesets [refer Talk 4]. To keep consistency in the system, one codepoint of WC must be unique among any locales and locale model must be a subset of the multilingual model – Global IOTMC model [3]. Also the multilingual system should be a single architecture to share all information by all components. Note that to satisfy above conditions, ISO specifications are not enough. And essential information to satisfy was discovered [later Talks] and handling ways of the information must be user definable.

Since informations behind characters were discovered, essential text manipulation functions for codeset/language/writing conventions were determined. Thus, a set of the functions can be provided as a *Widget*. Under mixing different writing direction characters, memory image and display image do not match, i.e., word order is changed. So, OM should have a role to return information about locations of a character and cursor. Therefore, displaying functions and returning such information should be text manipulation.

To use TMCs and TMC functions more easily, interactive association is essential – interpreter type programming languages must be provided. This means multilingual programming language should handle TMC as well as mb and WC. Note that TMC must be used to involve language informations.

4. Scope: Components and Functions

The essential components of the environment were developed as *Multilingual I/O TM/C System* [3]. By the research of generalizing text manipulation based on definition of character, clear definition of scope of the environment could be done. The multilingual environment should have Output [11], Input [1], Text Manipulation, Communication and Programming languages.

Output Module (OM) is called from basic drawing mechanism like X Window System [11]. OM determines an index of a font file and re-orders the words. Thus OM does not depend on glyph order in a font file.

Input Module (IM) could be generalized by the categorization of characters [1]. Adding flexible dictionaries to show candidates, IM could satisfy non-native speakers.

Text Manipulation Module has two parts; 1) WC and 2) TMCs. WC part can return information for line separation as essential information and can perform general text manipulation. Combination of the WC functions provided a Multilingual Text Widget. TMCs part can manipulate all data embedded in TMC bitfield.

Communication module (CM) has a role of encoding scheme conversion, because ISO 2022 has no default but locale model requires defaults. The conversion is essential to establish interprocess communication.

A Multilingual FORTH was developed for more advanced text processing. The FORTH can provide Multilingual LISP by its programming.

C compiler and all other libraries relating to character/text manipulation were replaced by using of the Meta Converter System. By those above, total multilingual computing environment was provided.

5. The Worldwide Multilingual Computing

The worldwide multilingual computing changes current ways of text processing and communication. By the functions provided, multilingual text formatting and printing can be daily use beyond codeset limitations. And international networking can be changed to true international communication by sharing all information. Furthermore, multilingual natural language processing can be started. Soon localization will be terminated and be replaced by multilingual applications.

References

[1] 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.

[2] Kataoka, Y., et al., 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, Proceedings of the 49th General Meeting of IPSJ, Vol. 3, September 1994, pp 299-300.

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

[4] Tanaka, T., et al., Multilingual I/O and Text Manipulation System(4): The Optimal Data Format Converter to/from MB/WC/TMC, Proceedings of the 49th General Meeting of IPSJ, Vol. 3, September 1994, pp 305-306.

[5] Kataoka, T. et al., Multilingual I/O and Text Manipulation System (3): Extracting the Essential Informations from World's Writing Scripts for Designing TMC and for the Generalizing Text Manipulation, Proceedings of the 49th General Meeting of IPSJ, Vol. 3, September 1994, pp 303-304.

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

[7] ISO/IEC 9899: 1990, Programming language C.

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

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

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

[11] Uezono, K. et al., Multilingual I/O and Text Manipulation System (2): The Structure of the Output Method Drawing the World's Writing Scripts beyond ISO 2022, Proceedings of the 49th General Meeting of IPSJ, Vol. 3, September 1994, pp 301-302.