

**Sequential Talks on the Multilingual Text Processing (4):**

1 Q - 4

**The Multilingual Input Method**

Hidejiro Daikokuyaf, Toshio Oyat, Tomoko Kataoka\*, Yutaka Kataoka\*,  
Kazutomo Uezono†, Dawa Yidemucaof, and Hiroyoshi Ohara†

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

**1.はじめに**

コンピュータネットワークの発展にしたがい、各コンピュータ上での国際化環境の整備が必須となっている。これに伴って、国際化テキスト処理のための必須機構としての入力機構が必要とされる。

入力機構が必要とされる理由は、一つには表意文字を表音文字からの変換[1]を行なうことによって入力する場合であって、この変換は文脈依存であり、最終的には利用者によって決定されなければならない。もう一つは、複数キーストロークによって一文字を決定する場合であり、入力機構がオートマトンにより状態を保持しながら文字の生成を行なうためである。

国際化した入力機構には、これに加えて意味のある単位を生成する機能が必須となる。意味のある単位とは次のように示される。(1)一文字を表すmbのバイト列。複数バイトや不定長バイトの文字集合がこれにあたり、一文字を決定するために必要なバイト列の途中で分割してはならない。(2)図形を表示できる文字の並び。複数の文字が連続していなければ表示出来ない場合であり、これは複数文字から一図形を決定する文字群と、ペルソアラビック文字のような位置依存性をもつ文字群に当てはまる。以上から、意味のある単位とはmb/WC/Glyph間の変換が正しくできる単位のことであることがわかる。入力機構はこの単位でアプリケーションとの通信ができる必要がある[2]。

さらに、実際に活用されるためには、ネイティブ/ノンネイティブ・スピーカーの違いによってそれぞれの利用者にとって容易な入力方法は異なるため、全ての文字が扱えるだけでなく、利用者の望む入力方法を選ばなくてはならない。

**2.入力の手法**

定義より国際化機構であるためには全ての文字を扱うことができなければならない。そのためには対象となる文字、文字集合を分析し、その各文字に適した入力方法をサポートしなければならない。また、この入力機構を扱う利用者について考えると、最も使用する可能性の高いネイティブスピーカーにとって扱い易いことは必然である。さらには、この国際化システムが語学教育に有効的に利用できることを含めて考えると、ノンネイティブスピーカーにとっても扱い易くしなければならない。

そこで、以上の要求を満たすために、単純にあるキーストロークに対応して文字を生成するだけでなく、以下のような入力が必要となる。

候補を表示して利用者が選択する方法は、一つの読みに対して複数の文字が該当することがあるような表意文字の入力に有効と考えられる。この候補選択の方法は次のような場合にも利用できる。タイ文字[3]には、図形上だけでは文字の区切りが特定できない時がある。特に文字に詳しくない利用者は文字の区切りを無視して、図形だけを入力しようとするのが考えられる。このとき、文字の区切り方が一意に定まらない状態になれば、考えられる文字の区切りの候補を表示し利用者に選択を促すことにより容易に入力を行なうことができるようになる。

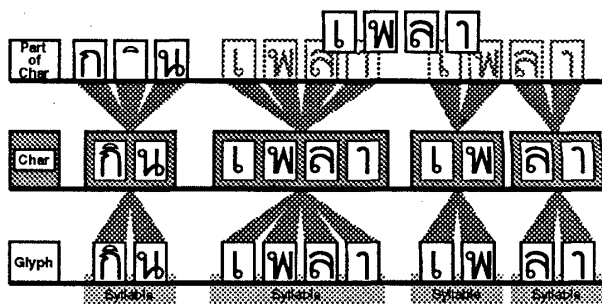


図1. タイ文字における文字の区切り

候補の選択については、アラビア文字において別な利用法がある。アラビア文字の場合、文字の区切りが同じであっても、文字と図形が一對一に対応する場合と、リガチャを行ない複数文字が一つの図形に対応する場合の両方がある。この場合にも、文字に詳しくない利用者にとって候補を表示し選択する入力が有効である。

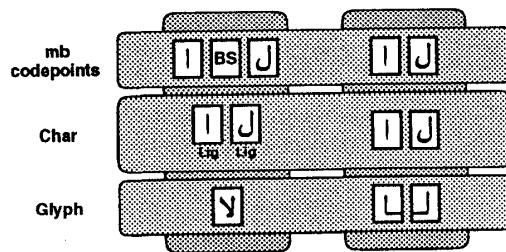


図2. アラビア文字におけるリガチャ

デヴァナガリ文字[4]に見られるパーミュテーションがある場合には、一つの言語内でもメモリエイメージとディスプレイイメージの順序が異なることがある。この時に、キーストロークの順序がどちらの順序で受け付けるかを選択できることが必要となる。

蒙古文字の場合、一つの文字に複数の変字体 (altanative) があり、その変字体が文脈依存によって決定されるため [5]、入力機構によって最終的には利用者が変字体の内から一つを決定しなければならない。さらには、デフォルトの入力で、各フォーム中で最も使用頻度の高い変字体が自動的に入力され、それ以外の変字体を入力したい時には必要に応じて利用者が変字体を選択することにより、入力時の手間を少なくすることができる。

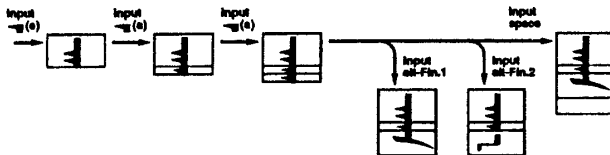


図3. 蒙古文字の変字体の入力例

キーアサインについても次のことが言える。

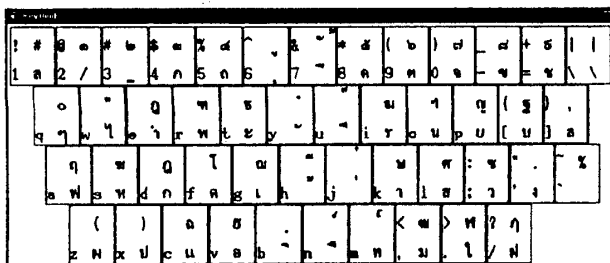


図4. キーアサインの表示例

任意のシフトキーを押した時にそれに対応して表示も変更する必要がある。蒙古文字のように違う文字が同じ図形を共有することがあるため、単に図形だけを表示するのではなく、文字を表す情報も表示しなければならない。また、キーアサインは文字とキーとの対応だけでなく、直前のキーのキャンセルやバッファをアプリケーションに転送するなどの機能を持つキーについても、ここで利用者に知らせる必要がある。このとき、文字を入力するためのキーについても、候補となっている文字の集合を限定する機能と考えることができ、そのため、機能とキーとの関係を表示すればいいことがわかる。

### 3. 入力機構の構造

入力機構の一般化は次のようにして実現する。入力機構に必要な基本的な機能を全て導出し、その機能をプログラマブルに組み合わせることにより、全ての機能を実現する。入力機構内部にマッチした時に上記の機能呼び出す、オートマトンを作成する。

入力機構は上記のオートマトンの要求だけでなく、入力機構全体構造上の機能として次のようなことが必要とされる。

IMM (Input Method Manager) は一つではなく、利用者が望んだ IMM による入力ができなければならない。そのため IMlib では、動的に任意の IMM とつなぎかえることのできる機能が必要とされる。こ

れは、全ての文字を扱うことの出来る IMM 上で全ての変換テーブルをオープンしたとすると大量にメモリを消費するが、大部分の利用者は全ての文字の入力を必要としないため、無駄にメモリを消費することとなるためである。同様に、LSC (Language Specific Converter) は一つではなく、利用者が望んだ LSC と交信できることが必要となる。

また、本来、蒙古文字は縦書きでしか表記することはできず、それに対してラテン文字などは横書きでしか表記することができない、そのため、IMP (Input Method Protocol) は縦書き、横書きの両方をサポートしなければならない。

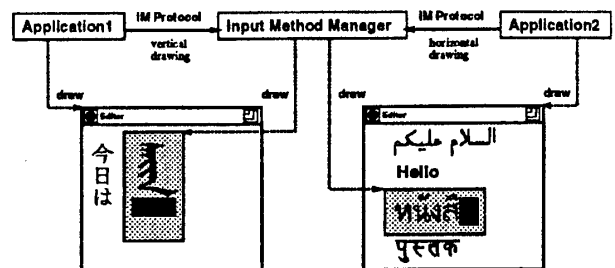


図5. IMP の縦書き、横書きのサポート

### 4. おわりに

各文字と各利用者に適した入力を単一の機構によって実現することが可能となり、入力機構の全体構造上で必要な機能についても分かった。

この入力機構は国際化テキストエディタが作成された時に有効に活用されることが考えられ、言語教育や図書館データベースの作成に役立つと考えられる。

### 参考文献

- [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] Daikokuya, H. et al., The Worldwide Multilingual Computing (6): Multilingual Text Interprocess Communication and Input Mechanism, Processing of the 51th General Meeting of IPSJ, Vol. 3, September 1995, pp 255-256.
- [3] TIS 620-2533 (1990), Thai Character Codes for Computers, Thai Industrial Standards Institute, Ministry of Industry, Thailand.
- [4] IS 13194: 1991, Indian Script Code for Information Interchange - ISCII, Bureau of Indian Standards, India
- [5] Kataoka, I, T. et al., Definition of the Mongolian Character Codesets Enabling Multilingual Text Manipulation, Special Interest Group on Computers and Humanities of IPSJ, Vol 29, January 1996, pp 61-66.