

日本語ワープロにおける入力モード自動変更方式

中 島 晃[†] 隈 井 裕 之[†] 松 本 通 顕^{††}

最近のワードプロセッサを初めとする OA 機器において、入力機能の操作性についての取組みが進んでいる。その中で、入力モード切り替えの操作に関しては、以前から問題意識が強い。ワードプロセッサを使った人ならば誰でも、「仮名」や「英数」といった入力モードの設定を誤っていたために、再び同じ文字を入力し直すという経験をしている。このような無駄を少しでも軽減するために、入力しようと思っている文字入力モードと、実際のモードが異なっている場合でも、システムが自動的にユーザの意図するモードを判断し、入力モードも正しいものに切り替えてくれると同時に、すでに入力された文字を自動的に意図するモードの文字に変換する「入力モード自動変更方式」を開発した。変更は仮名、ローマ字、英数の3モード間のものとし、入力文字1字ずつの確率頻度の累積をモードごとに比較することによってモード変更の判定基準とする試作システムを作成した。この試作システムについて、日本語・英語の文章計60文例の自動判定実験を行ったところ、13~20ストロークの文字入力力で93%の文例(56文例)が正しく自動判定をし、1文例が間違ったモードに変更し、変更誤りになった。残りの3文例は判定未了のものであるが、2~3文節の文章を入力すれば90%以上正しく変更されることになり、実用に耐えるものと判断する。

Automatic Change into Most Expected Input-Mode in Operating Japanese Wordprocessor

AKIRA NAKAJIMA,[†] HIROYUKI KUMAI[†] and MICHIAKI MATSUMOTO^{††}

Recently the investigation for input operation has been going on. We, user of Japanese wordprocessor, often feel trouble with setting or changing input-mode in making document. For, Japanese wordprocessor has non-single input-mode, that is "Kana", "Roman style", "Alphanumeric" and so on. We often have to reinput the characters in making mistake to set or change input-mode. We got the reasonable method that both the characters already inputted and input-mode can be changed into most expected ones even these cases. We calculate the sum of probability-frequency for each of input-modes every inputting character and if one of them exceeds others with threshold value, the changes of characters and input-mode is done. We measured the ability of automatic change by using 60 normal Japanese sentences. 93% sentences were correctly changed within 13 keyboard strokes. As for rest, they were not misjudge but non-judge.

1. はじめに

最近の、ワードプロセッサを初めとする OA 機器における日本語の入力は、文節間の意味関係を考慮した、AI かな漢字変換が主流になるなど、変換精度を向上する取組みが進んでいる^{1)~5)}。一方、入力に関する操作性にも検討が加えられている⁶⁾。その中で、入力モード切り替えに関しては、以前から問題意識が強い^{9), 11)}。もともと入力モードというのは、入力の効率をあげるために考案されたもので、代表的なモードと

しては、「ひらがな」、「カタカナ」、「英数」等がある。さらにこれらの文字を、「かな」モードか「ローマ字」モードのいずれかの方法で入力するのが一般的である。ワードプロセッサを使った人ならば誰でも、これらの入力モードの設定を誤っていたために、再び同じ文字を入力し直すという経験をしている。たとえば、

(1) 日本語の文章の途中で英字が混じっているとき、英字モードに切り替えて入力し、再び日本語の文章入力をするために、ひらがなモードに戻すのを忘れて入力を続けてしまう場合、

(2) 複数のユーザが同じワードプロセッサを使うとき、一方のユーザがまずローマ字モードで日本語を入力した後、もう一方のモードがかなモードで日本語を入力するとき、ローマ字モードに設定されてあるも

[†] (株)日立製作所映像メディア研究所第4部
Image & Media System Laboratory, Hitachi Ltd.
^{††} (株)日立製作所生活ソフト開発センター
Lifestyle Research Center, Hitachi Ltd.

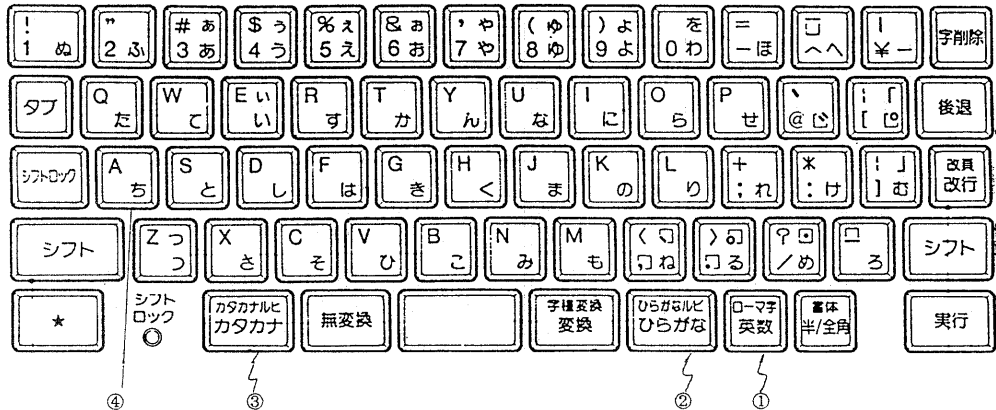


図1 日本語ワードプロセッサのキーボード
Fig. 1 Example of keyboard disposition of Japanese word-processor.

のをそのまま使用した場合、
などである。このような問題に対しては、入力の誤りに気付いた時点で、手で目的の入力モードに変更する機能がいくつか提案されている¹⁹⁾。しかし、字種変更の指示をユーザが行わなければならない、ユーザに余計な手間を強いるので、自動的に入力モードの設定ミスを修正する機能が求められる。しかし、これを実現する方法はまだ提案されていない。本稿では、文字を1字1字入力することによって自動的にユーザの意図する入力モードを推定する方法を提案する。

また、本機能がユーザの操作性に直接かかわることを念頭に置き、本機能の実現によってかえって操作等が煩雑にならないようにした。

2. 開発の背景

図1は日本語ワードプロセッサのキーボードを示したものである。図1のものは、文字キーの配列は英字、かなともJIS準拠であるが、親指シフト式、森田式、トロン仕様のものなど、JIS以外にもいくつか配列の異なるものもある^{8),9)}。しかし、入力モードを設定して文字を入力する点では共通している。たとえば、50音の1字1字を入力していく「かな入力モード」と、アルファベットを入力していく「ローマ字入力モード」の2通りの方法があり、図1の配列では、「★」と「④」の同時打鍵で切り替える。さらに「ひらがな」を打つか、「カタカナ」を打つかでそれぞれ、「②」、「③」を打鍵する必要がある。加えて、日本語文章中に混在する「英字・数字」を入力するときは、「④」を打ってから文字入力を開始しなければならない

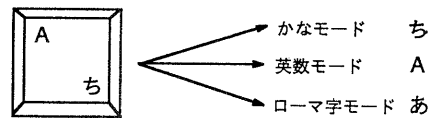


図2 入力モードごとの1キー打鍵結果
Fig. 2 Output character in each input mode.

い。すなわち、1つのキーに対し、入力のモード設定いかんで、3種類の文字が打てるようになっている。たとえば、図2に示すように、図1の「④」にあたるキーを打鍵すると、「かな」入力モードのときは「ち」、「ローマ字」モードのときは「あ」、「英数」モードのときは「A」が得られるのである。このように、ユーザはおおのこのモードを常に意識しなければならない、字種・入力モードを変更する場合にはその変更動作を行わなければならない。このため、モード設定ミスが多発し、ミスになった部分を抹消し、さらに入力のやり直しが必要となるため、ユーザがある種のストレスを感じる要因になっている⁷⁾。

このことは、実際の文章を入力してみるともっと明らかになる。図3は、日本語の文章を「かな」、「ローマ字」、「英数」のモード下で打鍵したときに表示される結果を示したものである。

(1)は、「きょうはてんきがよい」と「かな」入力で打つつもりで、各入力モードで打ってしまった結果、同様に(2)は「japanese」と英数で、また(3)は「じょうほうしより」とローマ字で打つつもりの場合を示している。いずれの場合も、入力モードの設定を誤ると意味不明の文字列が表示されてしまい、ユーザは困惑してしまう。

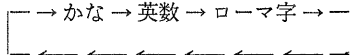
入力モードの設定を誤ったために入力をやり直すこ

とは、普通の人々が頻りに経験することなので、各方面で、そういうユーザの悩みを解消するための検討が重ねられている。現在、実用化されているのは、入力モードを、手動で変更するものであり、一部の専用ワードプロセッサ、パーソナルコンピュータで実施されている¹³⁾。手動によるモードの変更機能の操作手順は概ね次に示すようなものである。すなわち、文書の入力中に、入力モードの設定を誤ったことに気付いた時点で、ユーザが入力モードの変更を指示を行う。システムは、入力されている文字列を変更すべきモードのもとで入力された文字に読み替

え、結果を表示する。

変更の仕方は次のいずれかの方法が実施されている。

- (1) ユーザが入力モード変更のキーを打つごとに、たとえば、



のように、サイクリックにモードが変更された文字列に変更される。

- (2) まず、入力モードの変更を指示するキーを打ち、次に変更したいモードのモードキーを打つと、打たれた入力モードに読み替えた文字列に変更される。

以上のような方法ですでに入力された文字がユーザの意図する入力モードの文字に変更されるが、このあと入力される文字の入力モードについても、今までの入力モードを保持するものと、ユーザが変更したモードに移行するものがある。

しかし、この手動による入力モードの変更では、字種変更の指示をユーザが行わなければならない、ユーザに余計な手間を強いるものである。われわれは、この入力モード変更が、ユーザの操作を待たずに自動的に実現できる方式を検討し、次章で述べる累積出現頻度判定方式を開発した。

3. 入力モード自動変更方式

3.1 開発の前提条件

今回の開発に当たり、前提としたことを述べる。

(1)	G)	\$	F	W	Y	G	T	,)	E	
	き	よ	う	は	て	ん	き	か	^	よ	い	○
	かな											
	英数											X
	ローマ字											X
(2)	J	A	P	A	N	E	S	E				
	ま	ち	せ	ち	み	い	と	い				
	かな											X
	英数											○
	ローマ字											X
(3)	J	O	U	H	O	U	S	H	O	R	I	
	ま	ら	な	く	ら	な	と	く	ら	す	に	
	かな											X
	英数											X
	ローマ字											○

図3 モード設定誤りによる誤入力

Fig. 3 Example of mistyping caused from mis-setting of input-mode.

- (1) モードキーについて

漢字かな混じり文と英字等が混在する日本語の文章を作成する場合に特有な入力モードの設定や変更は、初心者ユーザにとっては負担である。これらの負担を解消するためのアプローチとしては、下記の2通りが考えられる。

①入力モードそのものをなくし、ユーザが自分の意図した文章を、自分の意図した入力モードのつもりで、自由に入力すると、システムが自動的にユーザの意図を汲み取って変換する。

②入力モードそのものはなくさず、正しく入力モードの設定や変更がなされているかをシステムが判断し、正しい入力モードに移行する。

①のように入力モードをなくせばユーザは自由に文章を入力することができるが、入力モードそのものが存在しないので、システムは、ユーザが入力した直後、どういう文字を打つつもりで入力したかの判定が困難であり、たとえば図3に示したように、3種類の入力文字列の表示をする必要がある。われわれは、まず、②の方法について、具体的な解決策を考えることにした。

- (2) 自動変更の対象とする入力モード

本研究の入力モード自動変更は、図4に示すように、下記の2つのケースのみを対象とした。

①ローマ字モードとかなモードの設定、変更誤り

②ひらがな、英数の入力モード設定、変更誤り

入力のモードは上記以外にも、カタカナモードや記

号モードがあるが、これらは本研究では、次に述べる理由から自動変更の対象から除外した。

①カタカナは、日本語の中で、主に外来語や特に強調したり読者の注意を引きたい語などに使うものであるが、本質的には日本語の 50 音を用いていることには変わりがなく、文字ごとの頻度などではひらがなとは区別のつきにくい性質がある。

そこで今回の開発では、ひらがな-カタカナ間については自動モード変更の対象とせず、4.2 節で述べるような方法でユーザの入力モード設定ミスに対処することにした。

②記号入力モードは、モードというよりも“記号機能”というユーザの感覚意識があり、入力するときには、特別に意識する。したがって、モードの設定、変更の間違いは比較的少ない。

3.2 入力モード自動変更のアルゴリズム (累積出現頻度判定方式)

たとえば、日本語の文章をすべてひらがなで書き下した場合、そのひらがなの系列を見ていくと、50 音、すなわち「あ」から「ん」までの文字は、同じ頻度で出現するわけではなく、「ひらがな」の系列の中に、他の文字に比べて高い頻度で発生する文字が存在する^{6),12)}。同様に、英字やローマ字にもそれぞれの系列の中に高い頻度で発生する文字が存在する。図 5 は、

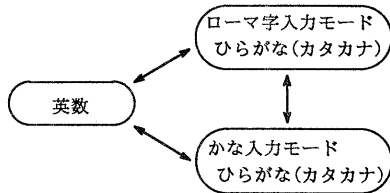


図 4 入力モード自動変更のモード間の関係
Fig. 4 Relation among input modes in the method of automatic change into most expected input mode.

出現頻度	可能性
① 母音の頻度大	“u-マ字入力”の可能性大
② 「あ、か、く」の頻度大	“仮名入力”の頻度大
③ 「s, t, d」の頻度大	“英数”の可能性大

図 5 文字の出現頻度とモードの可能性
Fig. 5 Example of the clue of deciding the expected input mode using appearance frequency of the character.

文字の出現頻度と入力モードの関係を表したもので、キーの系列に「a, i, u」などの母音が多く現れれば「ローマ字」モードの可能性が大きく、「あ、か、く」などの文字が多く現れれば「かな」モードの可能性が大きく¹⁰⁾、また、「s, t, d」などの文字が多く現れれば「英数」モードの可能性が大きいといえる。これをキーボードにあてはめると、「あるキーが入力される頻度はモードに固有の値を持つ」ということができる。

本方式の基本原理は、入力された文字列について、各モードごとに出現頻度の累積、すなわち「累積出現頻度」を計算し、最大の累積出現頻度と、次に大きいものとの差が所定の閾値より大きいかどうかを判定することにより、正しい入力モードを判定するというものである。

今、モードが k 番目の入力モード m_k に設定されているとし、ユーザが打ち込んだ文字の j ストローク目の文字 c_j に対する出現頻度を $p(m_k, c_j)$ とする。

各モードごとの文字の出現頻度 $p(m_k, c)$ は、次のようなデータ収集と、計算により得たものである。

(1) かな漢字変換の変換率の評価のために所有している、さまざまな分野の 90 文書 (かな読みに変換すると約 8 万文字) について、これをかな表記にした文書およびローマ字表記した文書を作り、おのおのについて各文字ごと (かなであれば「あ、い、…」、ローマ字であれば「A, B, …」) の出現頻度 (度数) を計算した。

また、英文処理評価用に所有している 30 文書 (英字換算で約 3 万文字) についても、英字の各文字ごとの出現頻度を計算した。

(2) 次に、かな、ローマ字、英字の各入力モードにおける最大頻度を与える文字の頻度が同一になるように各モードごとに頻度を正規化した。こうすることにより、入力モードごとの頻度の偏りが減少され、モード変更のチャンスが均等になる。

いま、 J ストロークまで打ち込んだときの k 番目の入力モードの出現頻度の累積 (累積出現頻度) P_{kJ} は、

$$P_{kJ} = \log \left[\prod_{j=1}^J p(m_k, c_j) \right] \quad (1)$$

で表される。ただし、式(1)は [] 全体の log をとっているため、 P_{kJ} は、各ストロークごとの出現頻度の log をとったものの和を計算していることに相当する。また、出現頻度の log をとっているため、log 自体が負にならないように、文字に対する出現頻度の値

は、一度も出現しなかった文字にも出現頻度 1 を与えてある。式(1)の場合、入力モードは m_k に設定されているが、ユーザがどのモードを意識して打ち込んだかわからないので、1つのキーに対して、可能性のあるモードすべてについての出現頻度の積を調べる必要がある。

たとえば、キーボード上の“A”の文字が打たれたとしても、それが漢字の“A”を打ったのか、ローマ字で“a”を打ったつもりなのか、また“A”のキートップに同時に刻印された“ち”をかなモードで打ったのかかわからないので、それぞれのモードでのこのキーに対する累積出現頻度が必要である。

今 K 個のモードのおのおのに対して、 j ($j=1, 2, \dots, J$) ストローク目までの累積出現頻度を計算する。

図6は、かなモードで「きょうはてんきがよい」と打てば正解となる文章について、「かな」「英数」「ローマ字」の3種のモードについての累積出現頻度を文字ストロークごとにプロットした例である。図6の縦軸に示した正規化出現頻度は、図の縦軸の目盛りのわかりやすさを示す便宜上、各モードの各文字の出現頻度すべてのうちの最大値が1になるように式(1)を正規化したものである。したがって、 j ストローク目までの正規化出現頻度 Q_{kj} には、

$$Q_{kj} \leq j \quad (2)$$

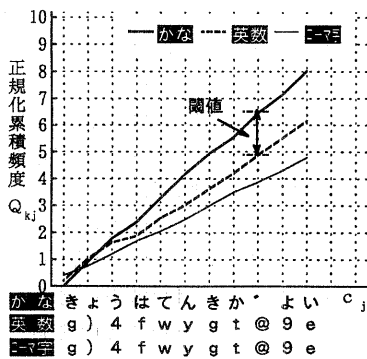


図6 累積出現頻度判定方式

Fig. 6 Input mode decision method using accumulation of appearance frequency.

という関係がある。図6を見てわかるように、ストロークが増加するにつれて、仮名モードについての累積出現頻度が、他の2モードについての累積出現頻度をだんだん引き離していくようになる。すなわち、この文字並びについては、ひらがなモードの頻度を増加させるような文字が並んでいる。

次に、 K 個のモードに対し、 J ストロークまで計算した累積出現頻度

$$P_{1j}, P_{2j}, \dots, P_{Kj}$$

に対し、閾値 P_{th} を適当に定め、

$$[P_{nj} - \max_{k \neq n} (P_{1j}, P_{2j}, \dots, P_{Kj})] > P_{th} \quad (3)$$

なる関係が成立したとき、入力モードを n に自動的に変更する。図6についていえば、9 ストローク目で「かな」モードに関する累積出現頻度が「英数」モードの累積出現頻度よりも閾値 P_{th} 以上大きくなった状態のことである。この場合に、たとえ入力モードが「英数」モードや「ローマ字」モードになっていても、「かな」モードであると判定するのである。以上の方式を、“累積出現頻度判定方式”と呼ぶことにする。

4. ワードプロセッサへのシステム組み込み

4.1 ユーザインタフェース

以上述べた、入力モードの自動変更機能を、ワードプロセッサのかな漢字変換システムに組み込んだ。

まず、ユーザはかな漢字変換を始める前に、自動モード変更の機能を利用するかどうかの選択をすることができる。そのユーザインタフェースを図7に示す。

図7は、ワードプロセッサにおいて、「拡張機能」というキーに続いて、「環境設定」を選択したときに表示される画面の例である。この画面例で、「モード変更」の部分が今回新たに付け加えられる項目であり、自動モード変更に関するユーザの選択画面である。

この選択は、図7に示すように、自動モード変更を利用しない(0:しない)場合と、利用するが、モード変更が必要になったときシステムが勝手に変更してし

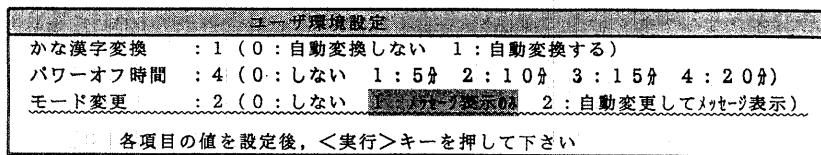


図7 ユーザ環境設定画面例

Fig. 7 Display example in user's initial setting.

まわず、ユーザにモード変更の必要性を知らせ、実際の変更はユーザに任せる(1:メッセージ表示のみ)場合と、自動変更を実行し、変更したことをユーザに知らせる(2:自動変更してメッセージ表示)場合の3通りをユーザインタフェースとして用意している。デフォルトとしては、図7に示すように、(1:メッセージ表示のみ)が設定されている。

次に、かな漢字変換の実行中に、自動モード変更が機能する様子を図8に示す。

図8に示すように、自動モード変更は、最初の4文字までは、機能させない。これは、評価実験の結果、最初の1~4文字程度の間は、3章で述べた自動変更のアルゴリズムを用いても変更誤りが生じるが、5文字以上になると急速に誤りが減少することが明らかになったためである。図8において、ユーザが入力した5文字目以降は、1字入力されるごとに3章で述べたアルゴリズムに従った判定を行い、閾値を超えた時点で、次に説明するようなメッセージを出力する。

図9は、図7でユーザの環境設定が、(1:メッセージ表示のみ)に設定されている場合の画面表示で、ローマ字で「hitatiseisakus」と入力した時点を示している。モード変更をするかどうかの判断は、「hitati…」の5文字目の「t」から1文字入力されるごとに行われ、図9の例では、「hitatiseisakus」まで打った時点で、モード変更をすべきであると判断されたことを示している。

表示メッセージは図9に示すとおりになっており、入力モードを手動で切り替えたい場合は、次節で述べるとおり、★+<変換>キーを打

てば、サイクリックにモードが変わる。ユーザは変更したいモードと文字列が表示されるまでこの操作を繰り返せばよい。

図10は、図8でユーザの環境設定が、(2:自動変更してメッセージ表示)に設定されている場合の画面表示であり、やはりローマ字で「hitatiseisakus」と入力した時点を示している。この場合は、「hitatiseisakus」まで打った時点で自動モード変更が実際に行われて、ひらがなモードに移行し、入力した文字もローマ字が打たれたと解釈され「ひたちせいさく」に変更されている。表示メッセージは、入力モード変更が行われたことを知らせ、その結果に異存がなければそのままひらがなモードの入力を続け、異存があれば手動モードで自分の意図するモードに変更することを指示している。

4.2 カタカナ-漢字変換機能

3.1節でも述べたように、カタカナはひらがなの区別が本質的に困難であるので、本研究では次のような方法でユーザの入力モード設定ミスに対処する。

①ユーザには、カタカナ語は相当数かな漢字変換辞書に用意されてある旨を説明し、できるだけひらがなで入力してもらう。

②従来、カタカナモードで入力した文字はかな漢字変換した場合もカタカナのままであったが、新たにカタカナ-漢字変換機能を搭載する。したがって、かな漢

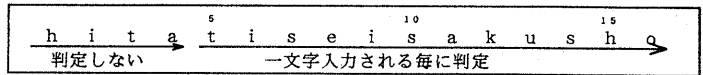


図8 自動モード変更機能の適用範囲
Fig. 8 Starting point of applying the method.

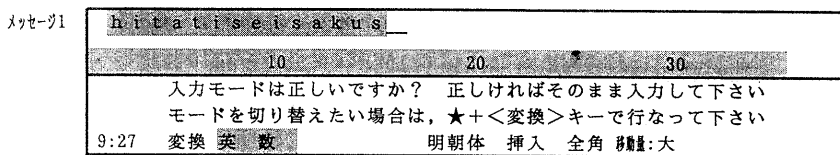


図9 メッセージ表示のみの場合の画面表示
Fig. 9 Example in the case of displaying only warning messages.

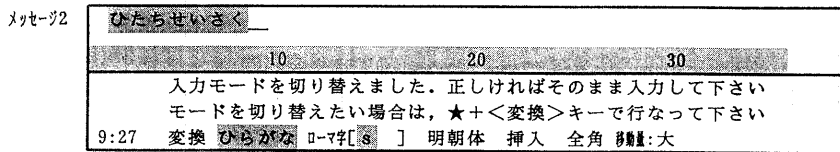


図10 自動モード変更時の画面表示
Fig. 10 Example in the case of automatic change of input mode.

字変換の読み中にあるひらがなとカタカナがかな漢字変換の対象になる。ただし、ユーザによっては、カタカナ辞書にも載っていないカタカナ語を故意に文章に入力したい者もいるので、カタカナ-漢字変換を行う場合を次のように制限する。

変換キーを押したときのモードがカタカナモードであること。そのとき、読みの最後の方から連続するカタカナ語のみをカタカナ-漢字変換の対象とする。

例：かな漢字変換の読みとしてユーザが次のように入力したとする。

「わたしはハナというアメリカウマレノニホンジンダ」

→カタカナ-漢字変換の対象になるのは、

「アメリカウマレノニホンジンダ」であり、「ハナ」は対象にならない。

ひらがなはどんな場合でもかな-漢字変換の対象となるので、上の例では、

「私はハナというアメリカ生まれの日本人だ」

という結果を得る。

4.3 自動モード変更誤り時の修正方法

入力モードの自動変更機能が誤ったときに、簡単に修正できる手段を備えることは、自動変更機能の精度の良し悪しにかかわらず必要である。われわれは、入力モードの手動変更機能を用いて簡単に自動変更誤り時の修正ができる操作を試作に組み込んだ。

図 11 は修正手順を説明する図である。

今、「X-mas をいろう」という文字列をローマ字で

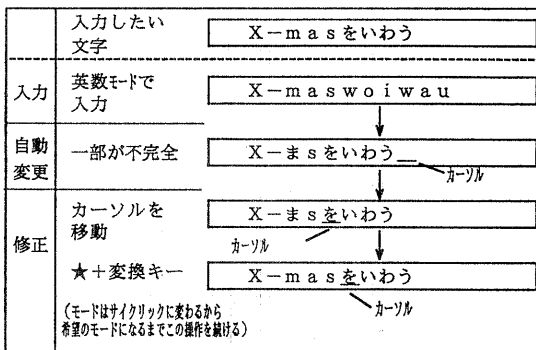


図 11 入力モード自動変更誤り時の修正手順
Fig. 11 Correcting process in the failure of this method.

入力しようとしている場合を考える。入力モードは「英数」に設定されているとすると、図 11 に示すように、ローマ字モードに設定されているならば、入力した文字がローマ字として成立する文字はただちにひらがなに変換されるが、設定されているモードが英数なので、英字がそのまま表示される。この入力が続けていくと、入力モードの自動変更機能でローマ字入力モードであると判断され、かなの読みに変更される。しかし、この例の場合は、最初の 5 文字は英字のまま残しておきたい文字である。このようなときは、図 11 に示すように、モードを別のものに変えたい部分の直後の文字 (図 11 では、「を」の部分) にカーソルを移動し、2 章で説明した手動モード変更機能を用いて簡単に修正できる。

具体的には、「★+変換キー」が手動モード変更を指示するキーに割り当てられ、これを打鍵すると、入力モードがサイクリックに変わり、カーソルよりも前の部分はそのモードで入力したとみなした文字列に変更される。したがってユーザは、目的のモードに変換されるまで「★+変換キー」を押せばよく、修正方法が簡単にユーザにわかりやすい操作手順になっている。

また、図 11 では 2 つのモードが混在する場合の修正方法を示したが、3 つ以上のモードが混在する場合にも同様な手順で修正できる。たとえば、

「kondonoX-maswoiwau」が自動変更されると、「こんどの X- mas をいろう」

となるので、「を」の部分にカーソルを移動して手動変更すると、

「kondonoX-mas をいろう」

となる。ついで、カーソルを「X」の部分に移動して手動変更をすれば、正しく

「こんどの X-mas をいろう」

を得ることができる。

5. 性能の評価

以上述べた「入力モード自動変更機能」について変更の精度に関する評価を行った。

5.1 性能評価の条件

ベンチマークの文章として、日本語の漢字かな混じり文を読み直したものの 40 文例と英語の文章 20 文例を用意し、40 文例を 2 つに分けて 1 つはかな読み直したものの、もう 1 つはローマ字読み直したものを用意した。すなわち、

①かな読み文 20 文例 (例：ほんじつはせいてん

- なり, …)
 - ②ローマ字読み文 20 文例 (例: honnjitsuha…)
 - ③英語文章 20 文例 (例: expectation…)
- 計 60 文例である。

5.2 評価結果

図 12 には上記 60 文例に対して行った入力モード自動判定の結果を示している。図 12 において、横軸はユーザが文字を入力したストローク数で、縦軸は、このストローク数を打った時点で、システムが各入力モードの文例に対し、その入力モードの文章であると正しく判断した文例数である。この結果を見ると、5 ストローク打った時点で全文例数の半分近くは判断がされていることがわかり、しかも各モードによる判断の差はあまりないことがわかる。図 13 は、図 12 のグラフのモードを統合し、縦軸にシステムがそのストロークまでに正しくモードを判断した文例の累積を全文例数に対する割合で示したものである。図 13 の閾値 0.5 に対する折線が図 12 のグラフの累積である。図 13 には同時に閾値 1.0 の場合の結果も示してある。閾値が高い場合、ストロークを多くしないとなかなか閾値に達しないため、自動変更のタイミングが後にず

れる。これに対し、閾値を小さくするとモード変更のエラーが多発する。閾値を 0.3~1.0 に変えて実験したところ、閾値 0.5 の場合がエラー率、ストロークのタイミングとも最も良好であった。図 13 の結果、6 ストロークまで入力すると全体の半数がモードを正しく判定されることがわかる。10 ストローク打てば 80%、13 ストロークでは 90% を超える。これらを実際の入力に当てはめてみると、ローマ字入力で 1 文節分のストロークが平均 8 ストローク、かな入力で 5 ストローク程度であるから、2~3 文節を入力すれば 90% 以上の場合、正しく入力モード自動変換が機能することになり、実用に耐えうると判断できる。また、20 ストロークまで入力した時点では、入力モードの変更誤りが 1 文例、このストローク入力時点ではまだ入力モードを判定できない (つまり、3 章で述べた累積出現頻度と他のモードの出現頻度との差が閾値を超えない) ものが 3 文例あった。

以上述べたように、間違った入力モードで文章を入力した場合の 90% の場合に自動変更が効くようにするためには、13 ストロークぐらひは続けて入力しなければならない。つまり、この入力モード自動変更方式は、こまめに 1 文節ごとに変換キーを押しながら文書作成をするユーザよりも、読みを長く、少なくとも 2~3 文節ずつまとめて変換するユーザに適した方式といえることができる。しかし、いずれのユーザの場合も、4 章で述べたように、ユーザが任意の時点で、「★+変換キー」を押せば、手動でモード変換をすることができることは言うまでもない。

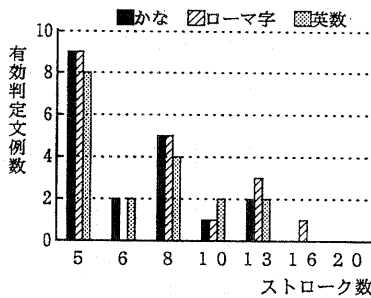


図 12 ストローク数-有効判定数の関係
Fig. 12 Relation between stroke counts and the number of change successfully.

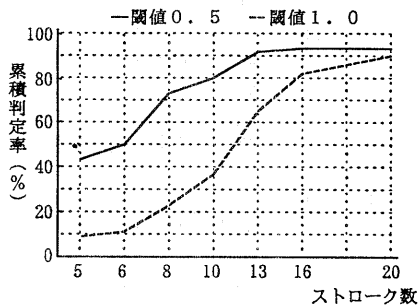


図 13 ストローク数対累積判定数の場合
Fig. 13 Stroke numbers vs. accumulated percentage of change successfully curve.

6. ユーザ操作性の評価

本機能の良し悪しは、ユーザの操作性の評価を抜きに考えられない。われわれは、一般ユーザに本機能を実際に体験してもらい、操作性に関する評価を行った。

6.1 ユーザ操作性評価の条件

人材派遣会社の大学生 8 名で JIS キーボードのワープロ経験者 (ローマ字入力 5 名、かな入力 3 名) に、モードレスが正しく動く文例と、誤って動く文例を適当に混合させたベンチマークを用いて使用させ、機能自体の魅力、操作性に対するユーザの意見等を抽出した。

6.2 評価結果

(1) 機能自体への反応 (魅力性)

今回モニタが入力した文例は、通常文章を入力す

るときよりも「誤って働く」ものを意図的にベンチマークとして用意したにもかかわらず、8人中7人までが「役に立つ」と答え、機能自体には肯定的であった。

(2) 機能性、操作性について

8名の被験者が本機能を使用したときの感想、改善要望点は次のようであった。

①自動モード変更について

変更対象	文字列、モード両方がよい	5人
	モードのみでよい	1人
	自動モード変更は不要	2人
英字／ひらがな混在の変更の実現		4人
メッセージをもっとわかりやすく		2人

②手動変更について

変更対象	文字列、モード両方がよい	2人
	文字列のみでよい	6人
	手動変更は不要	0人
手動変更対象文字列をわかりやすく		2人

この結果をまとめると、次のようになる。

- ①自動モード変更については、文字列、モード両方の変更を望んでいるが、手動変更は、文字列のみの変更を望んでいる。これは、自動モード変更が、文字列の入力中に利用されるのに対し、手動変更が、すでに入力された文字列の修正に利用されるためであると思われる。
- ②今回はサポートしなかった、2種以上の字種の混在文字列に対して自動変更の要求がある。

6. おわりに

パーソナルワードプロセッサのユーザにとって、煩雑で操作性を阻害する原因の1つになっている、入力モードの設定ミス、変更誤りをシステムが自動的に判断し、正しい入力モードに自動的に変更する「入力モード自動機能」を開発した。

- (1) 文字が入力されるごとに入力モードごとの累積確率頻度を計算し、閾値分の差が発生したときに自動的に入力モードを決定する、累積出現頻度判定方式を提案した。
- (2) カタカナについては、ひらがな-カタカナ変換の辞書を充実し、カタカナ-漢字変換の機能を新たに設けることによりモードの遷移を可能にし

た。

- (3) 読みの入力の任意の時点で使用できる入力モード手動変更機能を設けた。自動変更が誤った場合でもこの手動変更モードを用いることにより、ワンタッチで誤りの修正ができるようにした。
- (4) 60文例のベンチマークを用いて入力モード自動変更機能の評価を行ったところ、6ストロークまでで50%、10ストロークまでで80%、13ストロークまででは90%以上の文例が正しい自動変更をし、実用に耐えうることを確認した。

謝辞 本研究の機会を与えてくださった日立製作所OA事業部徳永赴プロダクト・マネージャ、多賀工場新井康彦工場長、システム開発研究所春名公一所長、遠藤裕英部長、ならびに、本研究を行うに当たりご指導いただいた映像メディア研究所松田泰昌主管研究員に感謝いたします。

参考文献

- 1) 日経エレクトロニクス編：日本語処理，日経エレクトロニクス・ブックス，pp. 63-98，日経マグロウヒル社 (1983).
- 2) 長尾 真監修：日本語情報処理，p. 353，電子通信学会 (1985).
- 3) Charniak, E. and Wilks, Y.: *Computational Semantics*, North-Holland (1976).
- 4) Fillmore, C. J. (田中春美ほか訳)：格文法の原理，三省堂 (1975).
- 5) 田中穂積：計算機による自然言語の意味処理に関する研究，電子技術総合研究所報告，797 (1979).
- 6) 加藤正隆：基礎暗号学Ⅱ-情報セキュリティのために，pp. 355-359，サイエンス社 (1989).
- 7) 石田晴久，木村 泉，安田寿明編：ワープロと日本語処理，bit 別冊 (1985).
- 8) 山本直三，河田 勉：日本語ワードプロセッサ，p. 201，オーム社 (1985).
- 9) 内藤 忍：ワープロ・タイピング修得法，p. 213，日本実業出版社 (1988).
- 10) 中山 剛，黒須正明：仮名文字鍵盤配列の評価と設計，情報処理学会日本語入力方式研究会報告，17-A 4 (1984).
- 11) 大野邦夫：ローマ字入力方式のモードに関する検討，情報処理学会日本語文書処理研究会報告，3-1 (1985).
- 12) 富樫雅文：中指シフト仮名文字配列「花」，情報処理学会ヒューマンインタフェース研究会報告，26-2 (1989).
- 13) 日経ホーム出版社：日経 TRENDY，pp. 20-21 (1992).

(平成4年3月23日受付)

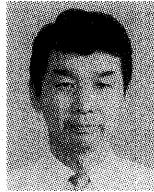
(平成5年11月11日採録)

**中島 晃**

昭和 20 年生。昭和 44 年東京工業大学理工学部電子物理工学科卒業。昭和 46 年同大学理工学研究科（電子物理工学専攻）修士課程修了。同年より（株）日立製作所中央研究所に勤務。現在、（株）日立製作所映像メディア研究所主任研究員。印刷漢字認識、音声合成、音声話者認識等の研究を経て、昭和 54 年から日本語ワープロの入力方式の研究・開発に従事。仮名漢字変換の変換アルゴリズム、ユーザインタフェースの研究に加え、ユーザがワープロなど OA 機器の入力をする際に犯すミス（入力モード設定誤り、タッチミスなど）を自動的に救済する機能の開発にも力を入れている。電子情報通信学会会員。

**隈井 裕之（正会員）**

1962 年生。1986 年北海道大学理学部化学科卒業。同年（株）日立製作所に入社し、以来、形態素解析、構文意味解析などを用いた日本語入力処理の研究開発に従事。現在、（株）日立製作所映像メディア研究所研究員。自然言語インターフェース、知識処理システムに興味を持つ。

**松本 通顕**

（株）日立製作所家電事業本部生活ソフト開発センター主任技師。生活ソフト開発センターでは「人・行動・くらし」の研究から新しい生活ソフトを見出し、商品コンセプトを創出することを目的としている。このため、Nature Pal と称する生活提案モニターを組織し、ヒューマンファクタ、ヒューマンインタフェース、ライフスタイル、環境のアメニティの四つの研究領域よりアプローチしている。現在、AV、情報家電を担当しており、ワープロは昭和 62 年より担当している。