

## 日本語ワープロ向け新打鍵レベル模型の 検証と応用

木村 泉、粕川正充  
(東京工業大学理学部)

**梗概** Cardらが導入した打鍵レベル模型(KL模型)は、テキストエディタなどのような会話型システムについては、利用者の操作所要時間をかなりの精度で予測する能力をもつが、変換型の日本語ワープロに原型のまま適用すると、しばしば予測値が大幅に過大になる。しかるに、もし打鍵レベル模型に「変換結果を見て、思った通りになっていれば先へ進む。」という、日本語ワープロの利用上よく出てくる状況に対応する新しい思考操作を追加すれば、上記の難点のない模型(新KL模型)が得られる。この既報の結果は、裏付けを主としてワープロソフト「松」についての実験から得ていた。本文では、新KL模型をさらに整理するとともに、これを別のワープロソフト「一太郎」による小実験に当てはめて検証し、その結果をもとに「一太郎」の連文節変換機能について考察する。

### Refinements and Verification of a Keystroke-Level Model for Japanese Word Processors

by Izumi Kimura, and Masaatsu Kasukawa  
Tokyo Institute of Technology, Department of Information Science

**Abstract.** The Keystroke-Level Model introduced by Card et al. predicts the user performance time with interactive systems such as text editors with considerable precision. However, it grossly overestimates if applied in verbatim to so-called conversion-type Japanese word processors. A modification of the model, presented previously, was based on experiments with a best-selling Japanese word processing software "Matsu". Here, the modified model is further refined, and tested by applying it to another similar software "Ichitaro". The fit is excellent, though it slightly tends to underestimate. An application of the model is discussed.

## 1. 新KL模型

Cardらの打鍵レベル模型(The Keystroke-Level Model —以下KL模型と略す)は、計算機システム利用者の作業時間を予測するための技法として著名である[1]。それは、熟練者が作業内容を完全に把握してから、誤りを犯すことなしに、所定の最適な方法で完遂した作業だけを考えるとという前提に立ち、テキストエディタなどでは、自乗平均根誤差にして±21%程度の精度をもつといわれている。

だが[2]で示したように、KL模型を変換型日本語ワープロに原型のまま適用すると、しばしば予測値が大幅に過大になる。そこで[2]では、NEC PC-9800シリーズ用ソフトウェア「松」[3]における実際の打鍵状況の観察に基づき、またCardらによる人の頭脳の一般的情報処理模型「モデル人間プロセッサ」[4]を参照して、KL模型の一つの改良案(以下新KL模型と呼ぶ)を導入した。それは、若干改良した上で示せば、次のようなものであった。

新KL模型では、利用者が作業に費やす時間を、図1に示すような種々の操作の所要時間の合計によって予測する。たとえばKは1文字の打鍵をあらわし、その所要時間(本文では操作名と同じKであらわす)は1分間テストなどの方法で利用者ごとに決定される。H(ホームिंग)は手を鍵盤のホームポジションと、どこかよそ(マウスなど)の間で移動させる操作であって、その所要時間(Hであらわす)はCardらにならって一応0.4秒としてある。P(指示)およびD(描線)は、それぞれマウスなどで画面上のものを指す動作、および線画を描く動作であるが、本文では指示デバイスについては論じないので、説明を省く。

MはCardらの「M操作」を受け継いだものである。M操作は「頭の中でふっと考える」という動作(思考操作—mental operation)を、打鍵やホームिंगと同格の一つの操作と見なしたものである。その平均所要時間(やはりMであらわす)としては、当面Cardらが実験で定めた原KL模型における値1.35秒をそのまま採用する。

さて新KL模型のおもな特徴は、思考操作をMのほかにもう2種、新しく導入したことである。すなわち思考内容が2種の可能性の間からの選択という単純なものである場合に使われるX操作、およびそれが3ないし4種の可能性の間からの選択というこれに次いで単純な場合に使われるY操作である。両操作の所要時間を操作名と同じ記号XおよびYであらわすことも上と同様である。所要時間XおよびYの値については、検討の余地が残っているが、ここでは一応それぞれ0.48秒、および0.55秒としておく。

図2は図1の思考操作M、XおよびYを、物理的操作の系列のどこに割り振るかを定める分配規則である。([2]の(旧)新KL模型からのおもな改良点は、この分配規則に含まれる。)この分配規則は(1)Mの分配、(2)Mの削減、

(3)並列事象の整理、の3部から成る。第1部では、思考が入り込むだろうと思われる箇所に仮にMを置く。実はそれらのMのうちには、長めのものもみじかめのものもある。その切り分けを第2部でおこなう。すなわちMのうち、0に近いと思われるものは省き、無視はできないが単純な思考しか含まず、比較的短時間で終わるとされるものはXないしYに置き換える。第3部では、同時に進行する事象の間の重複を省き、考えなくてよいものはかっこでくくる。最終的に得られた操作系列のうち、かっこつきのものは無視し、残余について図1に示す標準所要時間を合計したものを総作業時間の予測値とする。

図2において文字列とは、ひとつながりのものとして利用者に意識される打鍵系列(KおよびHの列)をいう。これはあまり厳密な定義ではないが、ともかく変換型日本語ワープロについていえば(1)文節ごとに変換をする場合の、その文節のかな表記、(2)多数の文節を一気に変換する場合の、その一気に変換される意味的にひとつながりとなった文のかな表記、(3)つねにまとめて打鍵される一連の機能キー(ファンクションキー)の連鎖、などは確かにそれに当たる。ただし矢印キーを不定個数打鍵してカーソルを画面上のどこかに移動する、というような場合には、打鍵個数が多ければ、ところどころで手を止めて画面を見なおすことが必要になるであろう。そのような場合には、矢印キーの打鍵はいくつかの文字列にわかれると考える。([1]の5.2節~5.3節にこれに関連した記述がある。)

図2第1部(c)におけるHとMの交換は総予測時間には無関係であるが、模型による予測を現実の操作系列に対応づけるときこのようにしてあった方が自然である。実際、ワープロソフト「松」による実作業のビデオ記録を観察してみたところ、この項目でどのような状況下では、ポーズが

	打鍵またはボタン操作	秒
K	最高のプロ	0.08
	上手なプロ	0.12
	平均的プロ	0.20
	平均的アマ	0.28
	でたらめな文字を打つ	0.50
H	複雑なコードを打つ	0.75
	ずぶ素人	1.20
	手をマウスまたは鍵盤へ	0.40
P	マウスでものを指す	1.10
D(nd,ld)	(くわしくはFitt's law) 線分を画く	$0.9 \times nd + 0.16 \times ld$
	(nd:区間、長さの総計ld)	
X	2種の可能性から選択	0.48
Y	3~4種の可能性から選択	0.55
M	X、Yの範囲を越える思考	1.35
R	計算機の応答時間	<実測>

図1 新KL模型(0.2版)の構成要素

人が身体を動かしてすること、および計算機のそれに対する応答を一行に書き並べたものから出発し、下記のように進む。

### 第1部 Mの分配

(a)文字列(引数文字列と固定文字列を問わず。ただし長い文字列が二つ以上の認知の単位にわかれている場合は別々に数える)の2字目以降、およびその終わりの区切りを構成するKを除き、それ以外のKの前にMを置く。

(b)文字列の構成文字でもその終わりを区切る打鍵でもないK(独立打鍵)の直前にもMを置く。

(c)ただし引き続くH、Kにおいて、Hがもつぱら速隔の特定のキーの打鍵(K)のためになされるものであるときはそのKの前のMをHの前に移す。

(d)それ以外のH(基準打鍵位置切り換えのためのH)の前にMを置く。

(e)指令を指定するPの前にMを置く。

### 第2部 Mの削減

(a)Mの直後の操作がMの直前の操作から完全に予測されるときMを省く。例 マウスでスクリーン上のものを指したあと、そのものを選択するためになされるマウスのボタンの操作に先立つM。

(b)Mの直後の操作がMの直前の操作から(標準的でない操作がなされる確率が無視できるぐらい小さいという意味で)ほぼ完全に予測されるとき、Mを省く。例 削除指令に対してつねに出される「よろしいですか?」というメッセージに対し、確認のために叩かれる改行キーに先立つM。

(c)Mの直後の操作が2個の可能性からの選択であるとき、MをXに変える。

(d)Mの直後の操作が3~4個の可能性からの選択であるとき、MをYに変える。

### 第3部 並列事象の整理

(a)Rに続く動作がRの完了を待たずにただちに開始されるときは、そのRをカッコでくくる。

(b)Kの直後にRがあるときは、そのKとRのうちの、所要時間の短い方をカッコでくくる。

(c)Kの直後にXまたはYがあるときは、そのKをカッコでくくる。

図2 M、X、Yの分配規則(0.2版)

手の移動(H)のあとではなく、むしろ前にあった。

ひとまず文字列の終わりを区切る打鍵(たとえば変換型のワープロにおいて、文節の自然な切れ目で打たれる変換キーの打鍵)に先立つMを考え、図2第2部の規則(a)または(b)の中でそれを取り除くという方針もあり得るが、ここでは(原KL模型との対応を見やすくする意味もあって)第1部(a)において、はじめからそういうものは除外することにした。

第2部の(c)および(d)に該当しないMはそのまま原KL模型のMであるものと見なすが、これは臨時的処置であり、再検討を要する。将来X、Yのほかにもう少し長めの、たとえばZといったものを導入する必要があるかも知れない。

第3部(c)項においてX、Yのみが言及され、Mが含まれないのは、当面原KL模型から借用した部分はなるべくそのままにしておこうとの意図からである。おそらく将来は、Mもここに含めて扱うことになるだろう。

なお図1、2の見出しに0.2版と付記してあるのは、これが[2]で示したものの改良に当たっており、しかし一方、明らかに最終版ではあり得ないことによる。

## 2. 新KL模型の適用例

ここで図2のいっそうくわしい説明をかねて、新KL模型をNEC PC-9800シリーズ用ワープロソフト「一太郎」[5]に適用してみよう。図3、4のタイムチャートは、「一太郎」で「高性能日本語処理システム」という句を打つてみたようすを示す。図3は、文節ごとに変換キーを打鍵する

というごく常識的な方法によったものである。この打鍵方法を「個別変換方式」と呼ぶ。図4では多数の文節について一気に変換を試み、誤変換が生じたときはあとから補正するという方法を取っている。この方法を「連文節変換方式」と呼ぶ。打鍵は「ローマ字漢字モード」によっている。ただし「処理」はヘボン風にshoriと打ち、にもかかわらず「システム」は訓令風にsisutemuと打つ、というようなことをしている。被験者は著者の一人(I.K.)である。

実験はKL模型の基本的仮定に合うように注意して実施した。すなわち「一太郎」に学習機能の停止を指示し、個別変換方式と連文節変換方式を交互に使用し、大きくつかえたときはデータを捨て、これを各方式について少なくとも3組ずつの、ある程度円滑に進行した作業の記録が得られるまで続けた。実際には途中つかえ、勘違いなどにより4組のデータを捨てる必要があった。連文節変換については、最後にもう1回、余分のデータを採取した。

使用機器は[2]に示したものと同じで、CPUはNEC PC-9801Eである。「一太郎」では辞書をハードディスクやいわゆるRAMディスクに置くこともできるが、ここではフロッピーディスクのみを用いた。タイミング記録は[6]の方法で採取し、時計割り込みのハンドラの走行時間に関する補正を加えた。[2]でも注意したように、[6]における計測の単位は10msecであるが、補正のため数字にmsec単位のはしたが出ている。以下、ミリ秒の位はつねに参考数字である。

打鍵タイミングはメークの時点(キーを押した時点)とブレークの時点(放した時点)を両方とも採取してあるが、

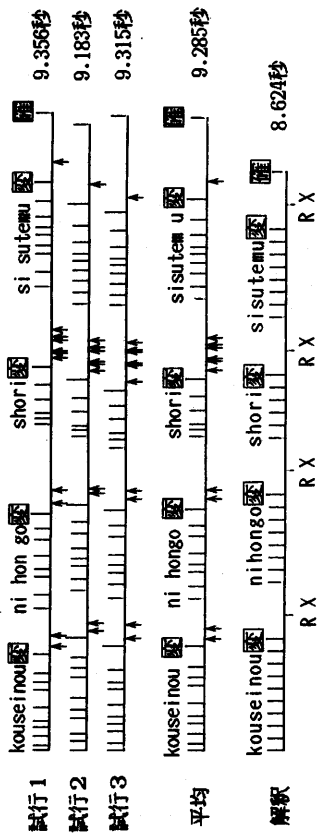
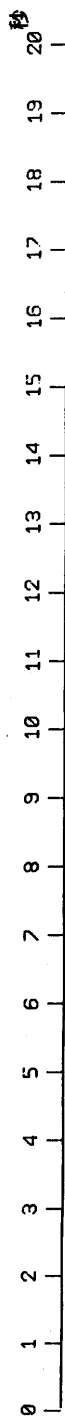


図3 「一太郎」における個別変換例

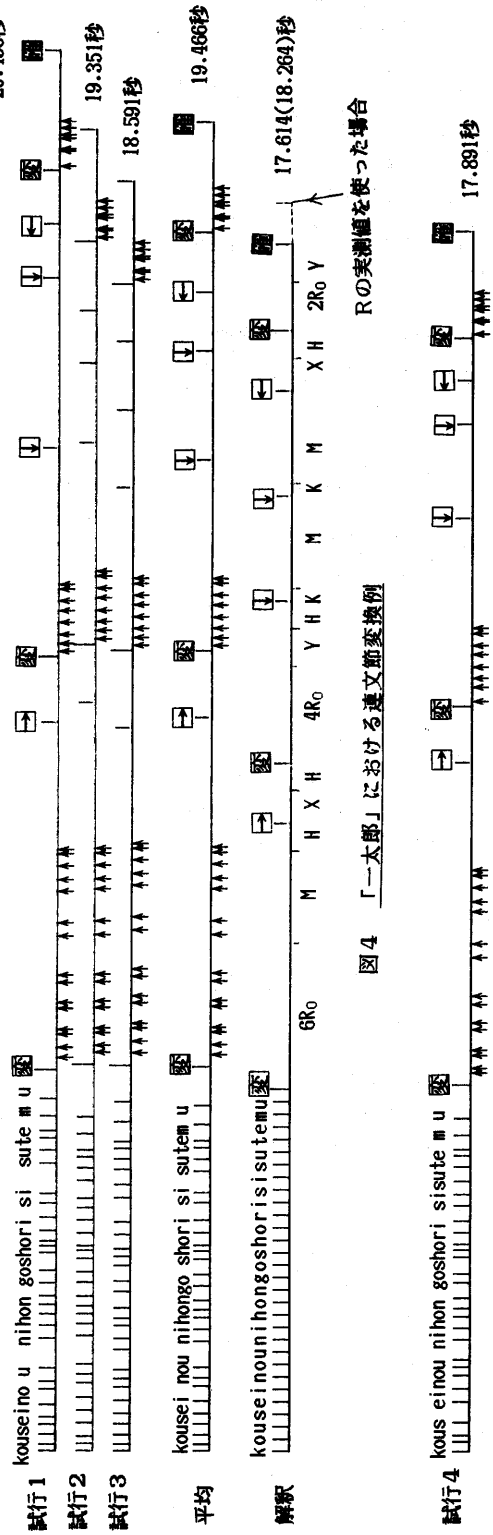


図4 「一太郎」における連文節変換例 Rの実測値を使った場合

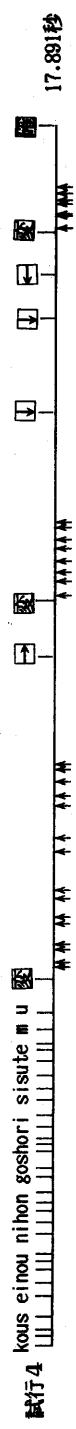


図5 4回目の連文節変換

	打 鍵	所要時間	画面表示
1	kouseinou	9K	----- こうせいのおう
2	[変]	(K)	
3	・	R	----- 高性能
4	nihongo	X 7K	----- 高性能 にほんご
5	[変]	(K)	
6	・	R	----- 高性能 日本語
7	shori	X 5K	----- 高性能日本語 しより
8	[変]	(K)	
9	・	R	----- 高性能日本語 処理
10	sesutemu	X 8K	----- 高性能日本語処理 しすてむ
11	[変]	(K)	
12	・	R	----- 高性能日本語処理 システム
13	[確]	X(K)	----- 高性能日本語処理システム

図6 「一太郎」による個別変換の詳細

図にはメークの時点のみ示した。これは[2]におけると同様、「一太郎」がキーのメークに反応してただちに動作をはじめることによる。図中「変」、「確」、[←]、[↓]および[→]を枠で囲ったものは、それぞれ変換キー（実はスペースバー）、確定キー（改行キー）、左矢印キー、下向き矢印キーおよび右矢印キーの打鍵を示す。ただし以下本文中および図6、7では、印字の都合上同じものをあらずのに枠の代わりにかぎカッコ[...]を用いた。

時間軸に下から入っている上向きの矢印はフロッピーディスクからの割り込みを示す。「一太郎」では画面の更新とフロッピーディスクの割り込みの時間的關係が必ずしも明確でない（たとえばシステムを立ち上げた直後には、画面が更新されたあとで余分のアクセスが起こることがある）が、ディスクの動作音から判断すると、図3、4に示す範囲では、一連のディスクアクセスの最後のひとつとほぼ同時に画面に変換結果が提示されると考えて差し支えないようである。

図3、4とも上から順に、3回の試行の生データ、それらの平均、および同じ事象の新KL模型による解釈を示している。なお上述のように連文節変換については図4に示すもののほか、最後にもう一組余分のデータを採取した。それを図5に示す。これを図4上部の3回分と合わせて見ると、連文節変換による場合には若干の練習効果が生じているように見える。実際、連文節変換による場合、利用者はおよそどのような操作が必要になりそうかを、事前には個別変換の場合ほどよく把握していないので、そのような効果が生じてもふしぎはない。

図6、7は図3、4の事象の、新KL模型による解釈の詳細を示す。やかましくいうと原KL模型、ひいては新KL模型には適用方法に関して二つのレベルがあり、これらの図に示すものは、その一つであるmethod-restricted予測[7]に当たっている。

まず図6について説明する。図2の第1部では「kousei

nou」、「nihongo」、「shori」および「sisutemu」が独立の(引数)文字列を構成し、その終わりの[変]（変換の指示）は図2でいう文字列の終わりの区切りとなっている。図2にしたがって、まず各文字列の冒頭にMを割り当てる。[変]にはMは割り当てない。最後の[確]は文字列の終わりを区切ってはいいない（独立の打鍵である）ので、その直前にMを割り当てる。ただし最初の文字列に先立つMは図では勘定に入れない。これは図3の実験において、総時間を最初の「kouseinou」の「k」の打鍵のメーク時点から最後の[確]のメーク時点までとしていることによる。

最初の一つを除く各文字列の打鍵は、直前の文字列についての変換がうまく行ったことを確認してからでなければはじめられない（打鍵を先にはじめてしまうことはできるが、変換結果が無条件に確定して補正ができない）ので、Mは計算機からの応答のあとに置かれる。Rは計算機の応答を示し、その長さ（それも一応Rで示す）としては本来は実測値を使うことに建て前である。しかしそれは一定値ではなく、状況によって変わってくる。ここでは簡単のため、図を値Rが毎回一定であるかのよう描いた。図3では、3回の試行についての各4回の応答の総時間はそれぞれ1.45秒、1.298秒、1.511秒、合計では4.259秒、応答1回当たりでは0.355秒であった。そこでここではR=0.355とした。図4の「打鍵」の欄の中心(・)はR1個を示す。

図にドミノの駒状のものを使って示したように、この作業におけるMは、いずれも二つの可能性の間からの選択（[変]または英字の打鍵、および[変]または[確]の打鍵）と考えられる。したがってそれらはXに置き換わる。最後の[確]の打鍵についてKにかっこがつけてあるのも、冒頭にMを置かなかつたのと同じ理由による。これは図2第3部におけるかっこづけとは関係がない。

総作業時間の予測値は、

$$29K + 4R + 4X = 29 \times 0.1822 + 4 \times 0.355 + 4 \times 0.48 = 8.624 \text{秒}$$

	打 鍵	所要時間	画面表示
1	kouseinou...	29K	
2	[変]	(K)	----- こうせいのうにほんごしよりしすてむ
3	.....	6R	
4	[→]	MH(K)	----- 構成の字に本語処理しすてむ
5	[変]	XH(K)	----- こうせいのうにほんごしよりしすてむ
6	.....	4R	
7	[↓]	YHK	----- 高性能日本語処理しすてむ
8	[↓]	MK	----- 高性能日本語処理しすてむ
9	[←]	M(K)	----- 高性能日本語処理しすてむ
10	[変]	XH(K)	----- 高性能日本語しよりしすてむ
11	..	2R	
12	[確]	Y(K)	----- 高性能日本語処理システム ----- 高性能日本語処理システム

図7 「一太郎」による連文節変換の詳細

となる。29はかつこのついてないKの総数である。Kとしては、図4の3回の試行における冒頭の「kouseinou ... sisutemu」の打鍵についての平均値を採用した（これは[2]における同じ被験者についての値0.219秒より小さい。I.K.は上達したらしい）。この総予測値は図3の平均作業時間9.285秒の7.1%減に当たる。

図6（および7）の「画面表示」の欄において、破線で囲った部分はワープロの画面上では黄色地、実線で囲った部分は青地、他は黒地に白で表示される。

次に図7について説明する。ここではまず「kouseinou ... sisutemu」を全部打ってしまったあと、一気に変換をしている。ところが、その変換結果は思わしいものではない。そこで左から順に補正をおこなう。

まず「高性能」とあるべきものが「構成の」で切れてしまっている。そこで[→]を叩いて、文節の切れ目を一つ右に移し、再度変換を指示すると「高性能日本語」までうまく行くが、その先が「処理しすてむ」となる。「システム」は辞書に載せてあるので、もし文節が「処理」の直後で切れればすべて丸くおさまるはずであるが、「処理し」とサ変動詞の活用形が拾われてしまうのでこうなる。

そこで[↓]（文節を1個確定させる働きをもつ）を2回叩いて「高性能日本語」までを確定させて「処理しすてむ」を残し、[←]を叩いて文節の切れ目を一つ左に移し、再度変換を指示する。これで思い通りのものができるので、あとは[確]を叩いてすべてを確定させればよい。

図7の画面表示欄において、下線が引いてあるのは、画面に黄色い下線がそのように引かれるものである。これはおそらく文節を左から順に区切って行ったとき、途中で行き詰まったことを示すものである。下線の部分以降では文字も黒地に黄色で表示される。

さて図2第1部の線に沿って、図7にMを割り振ってみよう。まず最初の「kouseinou...sisutemu」のところは全体を一つの文字列と見て差し支えない。明らかにこの程度

のものならば「ひとつながりのものとして利用者に意識」される。この文字列の先頭には1個のMが割り当てられるはずであるが、図6について述べたと同じ理由によってそれは図示していない。またこの文字列の直後の[変]は終わりの区切りをなしているので、Mの割り当ては受けない。

上記に続いてなされる打鍵[→]、[変]、... はいずれも独立打鍵であるから、その直前にMが配分される。利用者が熟練してくれば、状況を一目で判断して[→]+[変]とか[↓]+[↓]+[←]+[変]とかいった系列を一気に打鍵できるようになる可能性もあるが、ここでは利用者の熟練度はそこまでは行っていないと仮定した。事実、図3、4における被験者の技量は、その程度のものであった。

ところでPC-9800シリーズの鍵盤では、矢印キーは主鍵盤からかなり遠い場所にあるので、主鍵盤のキーと矢印キーの打鍵の間にはホーミング操作Hが必要である。それらのH（図7の4、5、7および10行目に1個ずつ必要）は、図2第1部(c)によってMの前に移され、HMという形からMHという形に変わる。

計算機からの応答に引き続き操作が、前者の完了を待っておこなわれなければならないことは図6の場合と同様である。応答時間は、図7の場合には複雑な処理を伴うのでかなり長い。図1では、応答時間については実測結果を使うことになっているが、ここでは一つの試みとして図6における平均値 $R=0.355$ 秒（ $R_0$ と記す）をもとに、概算値を出してみよう。まず「kouseinou ... sisutemu」全体を変換する最初の[変]に対しては、拾い出される文節が「構成の/字に/本/語/処理し/すてむ」の6個（最後の一つは分析不能なものとして扱われる）であることから、 $6R_0$ を見込もう。また2個目の、切れ目を調整した上での[変]については、できる文節が「高性能/日本語/処理し/すてむ」のように、全部で4個（うち最後の1個は認識不能）であることから $4R_0$ を、3個目の[変]については「処理/システム」のように2個の文節ができることから

ら2R<sub>0</sub>を見込む。総応答時間の見込み値は

$$(6+4+2) R_0 = 12R_0 = 4.26\text{秒}$$

である。ちなみに、図4の3回の試行での実際の総応答時間はそれぞれ4.987秒、4.815秒、4.916秒、平均では4.906秒であって、上記の予測より大きい。6R<sub>0</sub>とした部分では実際には9R<sub>0</sub>程度掛かっており、また4R<sub>0</sub>とした部分では3R<sub>0</sub>程度で済んでいる。図7の「打鍵」の欄の中心(・)1個は概算におけるR<sub>0</sub>1個を示している。

次に図2第2部のMの削減について考える。「構成の字に...」ができた直後(図7の4行目)には、図のドミノ印で示したように、次の5種の可能性が考えられる。

- [→] 文節の切れ目がまずい。みじかく取りすぎた。
- [←] 文節の切れ目がまずい。長く取りすぎた。
- [↓] 先頭の文節は適切だが、あとの方で訂正が必要。
- [変] 先頭の文節の切れかたはよいが、変換の候補がまずいので再変換が必要。

[確] 全部うまく行っているので、確定させてよい。実際にはこのうちから[→]が選ばれるわけであるが、5種の処置のどれが必要になるかは、変換の結果が出る瞬間まではわからない。すなわちここには5種、すなわち4種を超える選択の可能性があるが、図2第2部の(c)、(d)は適用されず、このMはMのままである。

次の[変](図の5行目)では、[→]を打ってみたらまだ不足で、もう1回[→]が必要と気づく、という可能性も十分考えられるので、場合は2種と数えた。したがってこのところのMはXに書き換わる。

第7行目の[↓]では、先頭の文節(高性能)の切れかたは一応利用者の思い通りになっているはずであり、したがって状況は第4行目において切れ目の補正に関する選択の可能性[←]および[→]を省いたものに当たっている。すなわち、可能性は3通りであり、MはYに書き換わる。

9行目では、二つ目の文節として切り出されたのは「日本語」であるということが、ここではじめて画面に表示される。それは一つ前の[↓]を叩く段階で見当がついてははずだから、ここは一も二もなく[↓]だ、とも考えられなくはないが、ワープロ利用者とはっさの判断の積み重ねによって行動して行くものであるから、その考えはやはり無理であろう。とすれば状況は4行目と同じと見られる。(ドミノ印のうち[確]のところは、8行目で「処理しすてむ」のところが思わしくないことをすでに知っている以上不要だ、という解釈もあり得るが、これもとっさの判断を積み重ねている利用者にはやはり無理であろう。)すなわち選択の可能性は5種、したがってMはMのままである。

10行目の[←]における状況は、ほとんど完全に4行目と同じであり、場合数は5、MはMのままである。ここでも、[確]は不要という考えかたがないではないが、上と同じ理由でそのままとする。

11行目の[変]は、5行目の[変]とほとんど同じで、場合数は2であり、MはXに書き換わる。

最後(12行目)の[確]は、図6の[確]と違って、もし「システム」のところがうまく変換されなければ、そのめんどうを([↓]を打って)見てやらなければならないという状況であるから、むしろ図7の7行目に近く、場合数は3となる。MはYに書き換わる。

このあと図2の第3部にしたがっていくつかのKをかつこでくるんだ結果が、図7の所要時間の欄に示されている。ただし最後の[確]についてのKのかつこは、図6におけると同様、図2の第3部とは関係なしにつけたものである。

かつこにくるまれていない操作全部について、所要時間を合計すれば総作業時間の予測値が得られる。それは、

$$\begin{aligned} & 31K + 12R + 4H + 3M + 2X + 2Y \\ & = 31 \times 0.1822 + 12 \times 0.355 + 4 \times 0.4 \\ & \quad + 3 \times 1.35 + 2 \times 0.48 + 2 \times 0.55 \\ & = 17.618\text{秒} \end{aligned}$$

であり、これは図4の3回の試行の平均値19.466秒の9.5%減に当たる。なお前述のようにこの予測値は、計算機の応答時間を過小評価している。上で12Rとあるところを実測値4.906秒で置き換えれば、18.264秒と変わり、実測値の6.2%減と、状況はやや改善される。図4の「予測」の行には、このようにしたときの状況が破線で付記してある。

### 3. 補足および討論

以上本文では、KL模型の[2]による拡張を、そこで考察の基礎に置いたワープロソフト「松」とは(特に連文節変換まわりにおいて)肌合いの違う「一太郎」に当てはめて検証し、その過程で模型を整理した。予測結果は、やや過小きみながら傾向をよくとらえており、まずまず満足できるものであった。気になるところがあるとすれば、図6、7の最終行の[確]で実測値が系統的に長いこと、および図7の9行目の[←]の実測値がMでなくXまたはYのように見えることであろう。

予測値が過小きみであるという点に関連して、次の事実がある。図3、4では模型による予測値を、3回の試行の平均と比較した。その代わりに3回の試行の各操作の経過時間について、通常のASCII文字の打鍵については引き続き平均を取るが、[変]、[→]などについては最小値を取るというようにしてみると、総作業時間の集約値として個別変換(図3)については8.993秒、連文節変換については17.845秒が得られる。これらに比べると、模型による予測値8.624秒および18.264秒(後者はRとして実測値を使用のもの)はそれぞれわずか4.1%減および2.3%増に過ぎない。この事実はKL模型の「熟練者が、作業内容を完全に把握してから...」という前提について、何ごとかを物語っている可能性が大きい。

本文の、実験および予測による検討結果から見ると、長い文字列を一気に変換できるようにするという、近ごろ多くのワープロ、ワープロソフトで試みられている行きかたには再考の余地がある。図7では誤変換が起こったために多大の操作時間が掛かっている(図6の場合の2倍以上)が、では誤変換がなく、ただちに正解が得られたとしたらどうか? その場合は、図7の第4行目の【→】の代わりに【確】が打たれ、そこで作業は終了することになろう(その【確】の前にはHは不要である)。総時間は、

$29K + R + M = 29 \times 0.1822 + 3.203 + 1.35 = 9.837$ 秒、と推定される。ただしR=3.203秒は「kouseinou ...」の第1回の変換における計算機の応答時間の、図4の3回の試行についての平均である。これは100%満足とまでは行かないが、まずまずの見積もりといえよう。

この9.837秒という値は、個別変換による作業時間の図6による推定値8.624秒よりずっと長い。すなわち、ここで考えている、辞書をフロッピーディスクに置いたという場合に関する限り、「一太郎」の連文節変換は変換誤りがあればもちろん、誤りがなくても、作業時間の単純比較に関する限り、個別変換と比べて損である。

では辞書をより高速の装置(ハードディスクやRAMディスク)に置いた場合にはどうか? やや乱暴ながら $R_0 = 0$ と仮定して考えてみよう。すると連文節変換における、新KL模型による予測値は、変換誤りがなければ

$$29K + M = 29 \times 0.1822 + 1.35 = 6.634 \text{秒}$$

図4の程度の誤りがあったときには、

$$\begin{aligned} & 31K + 4H + 3M + 2X + 2Y \\ &= 31 \times 0.1822 + 4 \times 0.4 + 3 \times 1.35 + 2 \times 0.48 + 2 \times 0.55 \\ &= 13.358 \text{秒} \end{aligned}$$

となる。一方この場合の個別変換に対する予測値は、

$$29K + 4X = 29 \times 0.1822 + 4 \times 0.48 = 7.204 \text{秒}$$

である。図4程度の誤変換が生ずる確率をpとすると、

$$6.634 \times (1-p) + 13.358 \times p$$

は $p > 0.0847$ では7.204より大きくなる。すなわち連文節変換が作業時間について得になるためには、大きな誤りが生じる確率は8.47%以内でなければならない。「一太郎」あたりの変換アルゴリズムでは、とてもそこまで行きそうもない。とすれば連文節変換の意義は、むしろ利用者がうっかり2個以上の文節を同時に変換しようとしてしまったときに救ってやれることもあること(および製作者側からいえば、複合語を辞書から落とせるので辞書ファイルが縮小できるかも知れないこと)に求めるべきではないだろうか?

さて本文では思考時操作XおよびYの所要時間をそれぞれ0.48秒および0.55秒とした。これは[2]では0.48秒および0.716秒としたものである。[2]ではXの値は実作業におけるタイミングのヒストグラムのピーク値から、Yの値はモデル人間プロセッサからの計算値を適宜かさ上げて、得

たものであった。前者は今回そのまま採用したが、後者については[2]において値0.48を出したと同様の実験を、新たにYについておこなって得たピーク値0.55秒を用いることにした。詳細については機会を改めて述べたい。

[2]ではXの値を導くのに、一つの発見的手法としてヒストグラムのピークに着目した。[8]ではそのヒストグラムを(多少修正の上)統計的に解析し、一つの解釈としてXはむしろ0.6秒程度とすべきではないか、と述べた。しかしその後さらに解析を進めた結果では、Xはやはり0.48秒でよかったようである。これについても改めて報告したい。

X、Yの値以外にも再検討すべきことはいくつかある。X、YのほかにはZといったものが必要かも知れない、ということはずでにいった。このほかHにも問題がある。PC-9800シリーズのキーボードでは、キーの近さ遠さに大幅なバラエティーがある。それをH一つでひっくくってしまうのは、多少無理のように思われる。また指示デバイス類に関する事項は、まだ原KL模型から取り込んできただけ、という形になっており、これからわれわれの環境の中で検証する必要がある。これらは今後の研究課題である。

ここでは[2]に引き続き、一人の被験者についての小規模な実験をもとに議論を進めた。しかしいつかは、大人数の被験者についての大きかりな実験も必要になるであろう。

**謝辞** この研究について科学研究費補助金(昭和60年度一般研究(C)第60580019号)の交付を受けた。

#### 参考文献

- [1] S.K.Card, T.P.Moran & A.Newell: The Keystroke-Level Model for User Performance Time with Interactive Systems, Comm.ACM, Vol.23, No.7 (July 1980), pp.396-410.
- [2] 木村 泉、柏川正充: 変換型日本語ワープロ向き打鍵レベル模型、第27回プログラミングシンポジウム報告集(Jan. 1986)、pp.95-106
- [3] 木村 泉、小沼千絵、柏川正充: 「松」とJWORDの比較論、コンピュータ・ソフトウェア、Vol.2, No.2 (Apr. 1985)、pp.420-444
- [4] S.K.Card, T.P.Moran and A.Newell: The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, New Jersey (1983).
- [5] 木村 泉: 一太郎の「住み込み」評価、日経バイト (Mar. 1986)、pp.185-201
- [6] 柏川正充、木村 泉: パーソナルコンピューター用打鍵データ収集プログラムとその応用、情報処理学会第30回全国大会、3G-8 (Mar. 1985)、pp.1645-1646
- [7] R.B.Allen and M.W.Scerbo: The Details of Command-Language Keystrokes, ACM Trans. O.A., Vol.1, No.2 (Apr. 1983)、pp.159-178.
- [8] 木村 泉、柏川正充: ワープロ利用者の思考時間分布、情報処理学会第32回全国大会、1K-4 (Mar. 1986)、pp.1801-1802