

# 漢字・日本語処理技術の発展： 日本語ワードプロセッサの 誕生とその歴史

天野 真家  
(株) 東芝 研究開発センター  
shinya.amano@toshiba.co.jp

森 健一  
東芝テック (株)  
kenichi\_mori@toshibatec.co.jp

## ④ ワードプロセッサ黎明期の概略史

日本語を誰もが簡単に書けるようにしたいという思想は明治期からあった。前島密の漢字廃止論、森有礼の英語化論、西周のローマ字化論、大槻文彦のかな文字論など、事例に事欠かない。いや、漢字から使いやすい平仮名を作った平安朝の時代からその思想はあったといってもよいだろう。このような思想と運動は連綿として続き、現在でも、(財)カナモジカイ<sup>☆1</sup>、(社)日本ローマ字会<sup>☆2</sup>が活動している。

電子計算機は、当初、日本語をローマ字、カナ文字で扱っていた。しかし、上記のような思想とは裏腹に、計算機に漢字を扱わせたいというニーズは、1960年代当時、すでに強固なものとなっていた。漢字は現在の常用漢字に限っても1,945字あり、人名用漢字284字を合わせると2,229字に及ぶ。この漢字をどのようにして、簡単、かつ、高速に入力できるか、より具体的には欧文タイプライタのようにタッチタイプで入力できるかは、大問題であった。当時、一般的な漢字の入力は、和文タイプライタのような全文字配列キーボード(図-1)を用いるか、あるいは12段もシフトがあるような漢字テレタイプライタを使うしかなかった。これらは、簡単、高速という要求を満たすには程遠いものであった<sup>☆3</sup>。

このような字単位入力のソリューションに対して、計



出典) 毎日新聞「10歳のワープロ社会」1989/8/22～

図-1 全文字配列和文入力機器の例

算機の機能を活用した言語学的ソリューションによる漢字入力法が「仮名漢字変換」であった。1960年代から70年代にかけて、九州大学、沖電気、NHK、NTT、大阪大学など<sup>☆4</sup>で仮名で書かれた文を入力し、それに言語学的解析を施し、仮名漢字混じり文に変換するアプローチが試みられていた。

言語学的アプローチの最大の課題は、変換率をいかに高くできるかにあり、それは日本語の機械文法<sup>☆5</sup>をどのように構成できるかに帰着した。

少なくとも、

- (1) 形態素解析のための機械文法の開発
  - (2) 同音語<sup>☆6</sup>に対処できる方法の創出
- の問題を解決しなくてはならない。

(1)は、仮名漢字変換の基本である。誤変換を極小にするには、この研究開発が不可欠である。本邦初の日本語ワードプロセッサであるJW-10の最大目標は、まず何よりも誤変換を減らすことであった。同音語の選択誤りは、タイピストにとって、後述する短期学習と長期学習により比較的容易な操作で解決できるが、誤変換は文字の削除操作と再入力が必要になるなどタイピストへの負担が非常に大きく、仮名漢字変換方式が受け入れられない要因になるからである。形態素解析は高精度仮名漢字変換のための基本技術であり、そのために機械用の辞書と文節の構造を記述した文法理論(形態論)を研究開発

☆1 1920年創立。http://www1.ocn.ne.jp/~kanamozi/

☆2 1885年羅馬字会設立。http://www.roomazi.org/

☆3 漢字入力の歴史は次に詳しい；浦城恒雄：漢字・日本語処理技術の発展－日本語の入出力と処理、情報処理、Vol.43, No.10 (Oct. 2002)。

☆4 巻末参考文献参照。

☆5 機械文法は、人間がすでに実践している言語活動を「説明する」ための従来の文法ではなく、それだけを手掛かりにして言語を「理解する」ための文法であり、立場がまったく異なる。

☆6 同音語とは、同音異義語を含むより広い概念として用いる。「飲む」と「呑む」など、日本語には同音同義(類義)異字が存在するからである。



図-2 ワードプロセッサの変遷

する必要があった。

(2)は、形態素解析のできる部分もあるが、同音異義語という名前が示すように、意味にかかわる部分が大きく、機械による完全な自動認識は困難である。JW-10では局所意味解析と名付けられた文節内意味解析と、学習機能によってこれに対処した。

1972年、筆者らは計算言語学的アプローチの予備的検討を開始。京都大学長尾研究室の協力を得て、日本語の構文解析の研究を始めた。翌1973年には、N.Chomskyの生成文法論、国立国語研究所発行の分類語彙表などを元に、意味論の問題を研究した。しかし、これらの理論の採用は、当時のハードウェアの低い性能、言語アーカイブ、オントロジーの未成熟などの理由から断念せざるを得なかった。

1974年、文節の形態素解析を基本とした仮名漢字変換方式の採用を決定すると同時に、従来の国語学的立場とは異なる計算言語学的な立場から、新たな国文法の研究・開発を本格的に開始した。同時に、辞書の構成法の研究・開発も進めた。計算言語学的国文法では、文法の土台となる品詞論から従来の文法論とは異なってくる。伝統文法の、名詞、動詞、接辞のような単純な分類による品詞論をとらないため、辞書の再編成が必要となる。対象となる語数は、既存の国語辞書の語数を参考にしても、

3万語以上は必要になる。さらに、新聞、雑誌に用いられる時事用語、オフィス独特の用語など一般の国語辞書には掲載されていない語も必要となる。

新しい文法体系と、その結果を用いて作られた新型辞書の開発、および大規模な実証実験、そこから生ずる新たな言語現象の発見、その解決のためのパラダイムの変更と、結果の文法へのフィードバックという過程が繰り返された。この結果、変換率<sup>☆7</sup>96.9%の実用に耐え得る言語処理システムが完成し、別途、開発していた、小型・低価格な漢字処理用ハードウェア、コンパクトなOS、スクリーンエディタと一体になって、1978年9月26日、JW-10として報道発表され、同時にデータショーに出展された。「JW」<sup>☆8</sup>は、Japanese Wordprocessorの頭文字語、「10」は、今後、7、5など、より小型、低価格の下位機種への展開と、また同時に20、30など、より高機能な上位機種への展開をも目指すという抱負を象徴して付けられた番号である。

図-2に、JW-10の試作機から始まって、1997年の20周年記念機までの簡単なワードプロセッサ(以下、WP)の変遷を示した。JW-10発表以降、電機、事務機器の製

☆7 多変換率と名付けた。同音語を変換候補に含む正解率である。

☆8 業務用には「TOSWORD」、個人用には「Rupo」という愛称が冠せられたのは後年のことであり、JW-10はまだ愛称を持たなかった。

入 力	入力モード： 漢字指定モード, 文節指定モード 同音語選択モード：一括選択モード, 逐次選択モード
編集校正機能	訂正, 挿入, 削除, 移動, 全文対象, タブ, インデント, アンダーライン, センタリング, 枠空け (図貼付領域), 切りばり, さし込み, 作表, 自動ページング, 保存, 呼出し, 禁則処理
文書記憶容量	本体内： 約 200 頁 (40 字× 40 行/頁の時) フロッピーディスク (8 インチ, 片面): 約 60 頁/枚 (40 字× 40 行/頁の時)
使用文字	文字種： 6,802 字 (JIS C6226) 字体： 明朝体

表 1 JW-10 のソフトウェア機能一覧

造会社から発売が相次いだ。1980 年代初期には、全文字配列鍵盤入力、音訓による単漢字入力、文法を持たず、辞書だけを持った仮名漢字変換入力など多彩な入力法による WP 群が発売されたが、1980 年代半ばまでにはこれらは姿を消し、JW-10 のように形態素解析を行う仮名漢字変換入力方式に収束していった。図-2 は、それらをも含む外観の変遷の歴史を代表したものである。

## ワードプロセッサのソフトウェア

表-1 に、JW-10 の機能一覧を示す。

WP のソフトウェアは、大きく、OS、仮名漢字変換、エディタ、ユーティリティに分かれる。OS は、JW-10 のために新規開発された。当初、TSS (Time Sharing System) で同時使用人数 4 人<sup>☆9</sup> を想定していたので、マルチタスク、マルチユーザをサポートする予定であったが、検討が進むにつれシングルユーザの形態となった。ユーティリティは、辞書への単語登録を含めた各種保守を行うための機能である。本稿では、WP の中核である仮名漢字変換部と、エディタ部について述べる。図-3 にソフトウェア構成を示す。

### 仮名漢字変換方式

JW-10 で最終的に採用された形態素解析を中心とする変換方式に至るまでに、統語解析、意味解析なども検討されたが、これらはすでに述べた理由で採用されなかった。いわゆるべた入力もアイデア段階で、断念したものの 1 つである。JW-10 の主記憶は、最大で 64KB であった。この中に、OS、仮名漢字変換、エディタを入れる

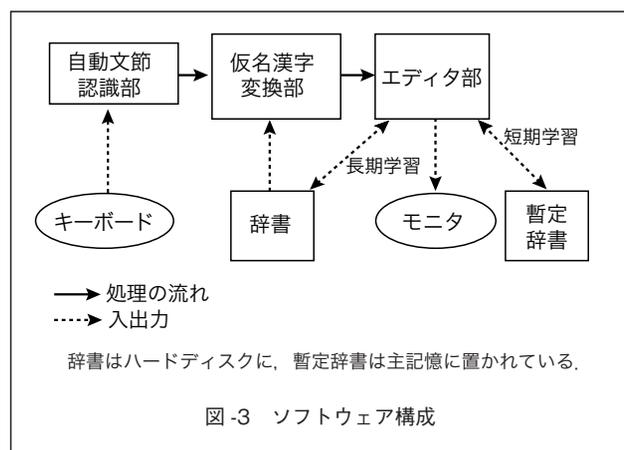


図-3 ソフトウェア構成

と、オーバーレイ<sup>☆10</sup>構成にしてなお、文書バッファでさえ 2KB しか取れなくなった。べた解析を行うためには文節の切れ目を推定しなければならないが、この候補は組合せの爆発を起こすので思考実験レベルで断念した。

断念した大きな理由の 1 つには、上記の技術的理由だけでなく、現実的理由があった。当時、2,000 万円はするというハードウェアの価格をどこまで下げられるかが問題であったが、個人が買えるようなものではないことは容易に想像がつく。最初のターゲットはオフィスである。この場合、原稿がある利用法が多いたらうと想定された。タイピストは依頼された原稿に忠実である必要がある。漢字は漢字、仮名は仮名のままタイプしなければならない。このため、後述する「漢字指定モード」という入力モードが用意された。この方法は、ちょうど、英文タイプライタに大文字、小文字のシフトがあるように、漢字、平仮名、片仮名などのシフトを持たせたものである。漢字指定モードでは、タイピストは文節という単位を意識せずに機械的にタイプすればよい<sup>☆11</sup>。自動文節認識部が擬似文節として自動認識して取り出し、仮名漢字変換部の中にある文節解析部で確定する。シフト操作はあるもののタイピストはべた文として入力することができた。これが、現在の個人用の WP における、仮名になるか、漢字になるかは機械任せのべた文入力のような

☆9 長期学習の頻度項目数が 4 であるのは、この影響である。  
 ☆10 Overlay：主記憶の同一領域を複数のアプリケーションで共用するメモリ管理方式。OS が仮想記憶をサポートしている現在では過去の技術である。  
 ☆11 「文節」というような専門用語を一般の使用者に理解させる困難を回避できることも、このモードの存在意義であり、右手と左手の親指操作だけで、容易に漢字と平仮名のシフトが可能であった (図-4)。



図-4 JW-10のキーボード

方法を採用しなかった大きな理由の1つである。

### 自動文節認識

JW-10では、2種の入力方式が用意された。漢字指定モードと、文節指定モードである(図-4)。漢字指定モードは、漢字シフトキーのような各文字種用のシフトキーを押して、文字種を指定するものである。文節指定モードは現在通常に行われている入力法で、文節の切れ目ごとに文節キーを押す方式である。漢字シフトキーと文節キーは同一のキーになっている。図-4にJW-10のキーボードを示す。

文節指定モードで入力された文字列には、タイピストによって入力された文節の切れ目を示すコードが入っている。自動文節認識部はこのコードで文節を認識して出力する。一方、漢字指定モードで入力された文字列は、タイピストによって文節に切り分けられていない。自動文節認識部は、この連続する文字列から、内蔵されたアルゴリズムによって擬似文節を抽出する。漢字、平仮名、片仮名、英字、記号などの文字種情報と、付属語辞書情報から、このアルゴリズムによって抽出される単位を擬似文節と呼ぶ。

自動文節認識部は、「[ひ]の[まる]を」<sup>☆12</sup>、「[ず]がい[こつ]に」などのように漢字指定モードで入力された文節がまぜ書きされていて、間に平仮名がある場合には誤認識して、前者では、「[ひ]の」、「[まる]を」、後者では、「[ず]がい」、「[こつ]に」と2文節にする。「[ひ]の[まる]」は、1語なのだが、本来は句であるので、言

語处理的には、誤りにはならない。ただ、「日の丸」に限定できず、「火の丸」なども候補に挙がるという悪い効果は起きる。「[ず]がい[こつ]に」も同様な問題を引き起こす。しかし、このような例はむしろ稀なので、全体的な変換率は文節指定モード入力より変換率が良くなる<sup>☆13</sup>。

### 2層型仮名漢字変換方式

仮名漢字変換部は2層になっている。第2層には、橋本文法の文節を処理する仮名漢字変換エンジンと固有名词変換エンジンを置き、第1層には、橋本文法を事実上逸脱する言語運用<sup>☆14</sup>的現象に対処する処理部を置いた。この第1層部を局所意味処理部と呼んだ。この2層構成がJW-10の仮名漢字変換アルゴリズムの最大の特徴であり、その変換率はこの構成に負っている。

橋本文法によれば、文節の定義は、下記ようになる。

文節=自立語+(付属語\*)

\*: 繰り返しを許す

(,): 省略を許す

<sup>☆12</sup> “[”は漢字-in, ”]”は漢字-outを示す。実際に、“[”や,”]”を入力するわけではなく、漢字シフト操作を可視化したものであり、また、内部表現でもある。

<sup>☆13</sup> 「火とは」、「人は」のような同音文節がなくなる。

<sup>☆14</sup> N.Chomskyの“Performance”を意味する。ここでは、文字通り「実践的運用」と理解されたい。

ここに、自立語、付属語という要素は、言語運用的立場では辞書で定義される実存在であるが、言語能力<sup>☆15</sup>的には、これらの要素は概念的存在にすぎない。後者の立場では、「第123回定期株主総会が」という文節は、「第123回定期株主総会」という自立語と、「が」という付属語から構成され、橋本文法で解決されるが、運用的には、このような処理は困難である。「第123回定期株主総会」の中には、「123」という変数が含まれる。これを辞書項目にしようとすれば、「第1回」から「第∞回」<sup>☆16</sup>までを登録することになり、困難が伴う。このため、上記、橋本文法による文節の定義をより具体的に拡大しなければならず、その解釈系が第1層となる。

JW-10では、次の、機械文法用文節の定義を用いた。

■普通文節■

普通文節の定義は、下記ようになる：

普通文節 = {(接頭辞) 自立語 (接尾辞)}\* + (付属語\*)  
ここに、

- \*：1回以上の繰り返しを許す
- (, )：省略を許す
- {, }：グルーピングを示す

この定義に含まれる接頭辞、自立語、接尾辞、付属語は、上記の例のような概念的存在ではなく、辞書に含まれる現実の存在である。「\*」の存在は、複合名詞の自動合成を行っていることを示している。複合名詞の切れ目は、通常、人間にとって認識が難しいものではない。認識が難しい、あるいはできないということは、意味が分からないということの意味する。社内文書や、自分で書く文書でそのようなことは起きないだろう。しかし、うっかり切らずにタイプしてしまうことはあるだろう。このことを考慮して、JW-10では、これをもサポートした。

また、この定義から分かるように、JW-10では接頭辞、接尾辞は、それぞれ、「語の末尾には置かれることができない接辞」、「語の先頭には置かれることができない接辞」として定義されている。

片仮名および英字からなる単語もこの定義を満足するが、JW-10では原則として辞書に登録されない。このため、これらの文字種を含む文節は、図-5に示すように、平

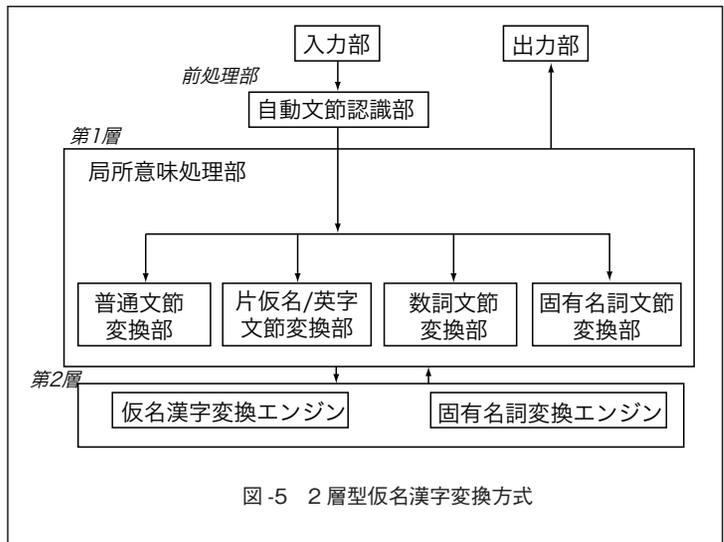


図-5 2層型仮名漢字変換方式

仮名(漢字シフトを含む)だけで入力された文節とは異なる処理の流れになっている。なお、片仮名、英字、英記号は、シフトによってそれらのコードが直接キーボードから発生されるが、漢字シフトキーは、漢字-in、漢字-outコードとして表現される。

第1層部は、これらの情報から文字種を判断して、その文字種に適した処理を施す。一例を挙げる。

「[[ひらかな] カタカナ [ひらかな] は」

という形式の文節が入力された場合、先頭の「[[ひらかな]」は、まず接頭辞とみなされ、「この文字列を、接頭辞として処理せよ」というモードで第2層の変換エンジンに引き渡す。結果が成功すれば、この部分は、接頭辞となり、変換失敗が返されれば、再度、「この文字列を、名詞として処理せよ」というモードで変換エンジンに引き渡す。成功すればその結果を、失敗すれば平仮名を仮名漢字変換の結果とする。

■数詞文節■

数詞文節は、下記のように定義した。

数詞文節 = (前置助数詞) 数字 (後置助数詞) (助数詞)  
(付属語\*)

従来の国文法では、「助数詞」があるだけであったが、機械用にはこの定義では不十分であり、JW-10では3種の接辞に分けた。前置助数詞は、数字の直前にのみ置かれる接辞、後置助数詞は、数字の直後にのみ置かれる接辞、助数詞は、数字の直後、および後置助数詞の直後に

<sup>☆15</sup> N.Chomskyの“Competence”を意味する。これはプラトンのアイデア類似の概念であり、計算言語学は、前述の“Performance”との乖離に常に悩まされている。ここでは、「理論的能力」と理解されたい。

<sup>☆16</sup> 言うまでもなく、運用的には適当な数でやめることになる。



置かれる接辞とした。

「だい」は、前置助数詞としては、「第」があり、「後置助数詞」としては「台」がある。また、助数詞は、「3回程度」の「程度」に対応するためのものである。さらに、前置助数詞と後置助数詞の共起にも共起表を持って対応した。JW-10では、1980年代中頃になってようやく実装され始めた「共起」を、すでに部分的に実装していた。なお、漢数字には、漢字シフトの中で、数字を入力することで対応している。「[[だい12<sup>☆18</sup>かい]]」は「第十二回」と変換され、「[[だい]12[かい]]」は「第12回」と変換される。

#### ■固有名詞文節■

固有名詞文節には、普通名詞文節とは異なった構造をもたせ、県市町村のような行政単位の構造を認識させるようにした。そのため、JW-10は、固有名詞シフトキーを備えている。これは、ひとえに同音語を減らすための工夫であった。固有名詞そのものが、普通名詞と同音語になることをなくすと同時に、接辞の同音語も削減することを目的とした。

『[[かながわけん かわさきし]]<sup>☆19</sup>』は、「神奈川県川崎市」に一意的に変換されるが、『[[かわさきし]]』は、「川崎氏」、「川崎市」、「河崎氏」に変換される。

固有名詞は、組織名などでは普通名詞からできているものがかなりの頻度で存在する。そこで、局所意味処理部は、固有名詞文節でも固有名詞処理エンジンが変換失敗を返してきたときには、その部分を通常文節の仮名漢字変換エンジンに「名詞文節として変換する」モードで引き渡し、結果を合成した。国際証券、三井造船、住友機械工業<sup>☆20</sup>、など、この例は多い。

#### ■人間-機械系としての同音語への対処■

日本語入力を仮名漢字変換という単体技術ではなく、WPとして、人間-機械系として見ると、仮名漢字変換単独では解決が困難であった問題が、総合的に解決できる。このことに気付き、人間と機械が良きパートナーとして機能するような仕組みを創案し、実装したことがJW-10の大きな特徴であった。具体的には、暫定辞書を用いた短期学習と、いくつかの分野、あるいは使用者の用語使用傾向を自動的に学習し辞書に反映する長期学習が挙げられる。

#### ■短期学習■

同じ文書内では、使われる同音語には偏りがあり、同じ語が多く使われるであろうという仮定で設けられた機能である。「こうしょう」は、小学館国語大辞典では61の同音語を持つ。「海軍工廠」の「工廠」は、その23番目に位置する。タイピストは、「工廠」が現れるたびに22回もの次候補キーを叩かなければならない。

エディタは、暫定辞書と呼ばれるスタックを持ち、タイピストが選択した同音語をここに記憶していた。エディタは、仮名漢字変換部から受け取った変換結果の中に同音語のリストがあると、この暫定辞書を調べ、その中に存在する最初の語を、リストの先頭に移動した(図-6)。モニタ画面には先頭の語だけを表示し、かつ、他の同音語を背後に持っていることを示すためにその部分をブリンクさせた。暫定辞書の効果の範囲は一文書であった。文書を閉じると、暫定辞書はクリアされた。これが短期学習という名の由来である。

#### ■長期学習■

長期学習は、辞書の各語に、同音語の選択操作を反映した使用頻度を記憶する4個分の欄を設けて4人、あるいは4分野の同音語の使用傾向に対応しようとしたものである。WPを使用するときに、登録されている分野のどれを使うかを指定すれば、辞書のこれまでの頻度に応じた順で同音語が表示される。1位に表示された場合、選択操作は必要ない。

JW-10では、現在の多くのWPが採用している同音語選択方式を、逐次選択方式と呼んでいた(図-4)。このモードでは、同音語を選択せずに次の入力を行うと、先頭にある語が自動的に選択されたが、もう1つのモードである一括選択モードで入力すれば、同音語は選択されず保持されたままであった。これは、同音語の選択のために思考が中断されることがないようにする配慮であった。このモードでは、同音語は、任意の時点で選択できる。そのために、同音語の存在を示すため同音語はブリンクを続けた。

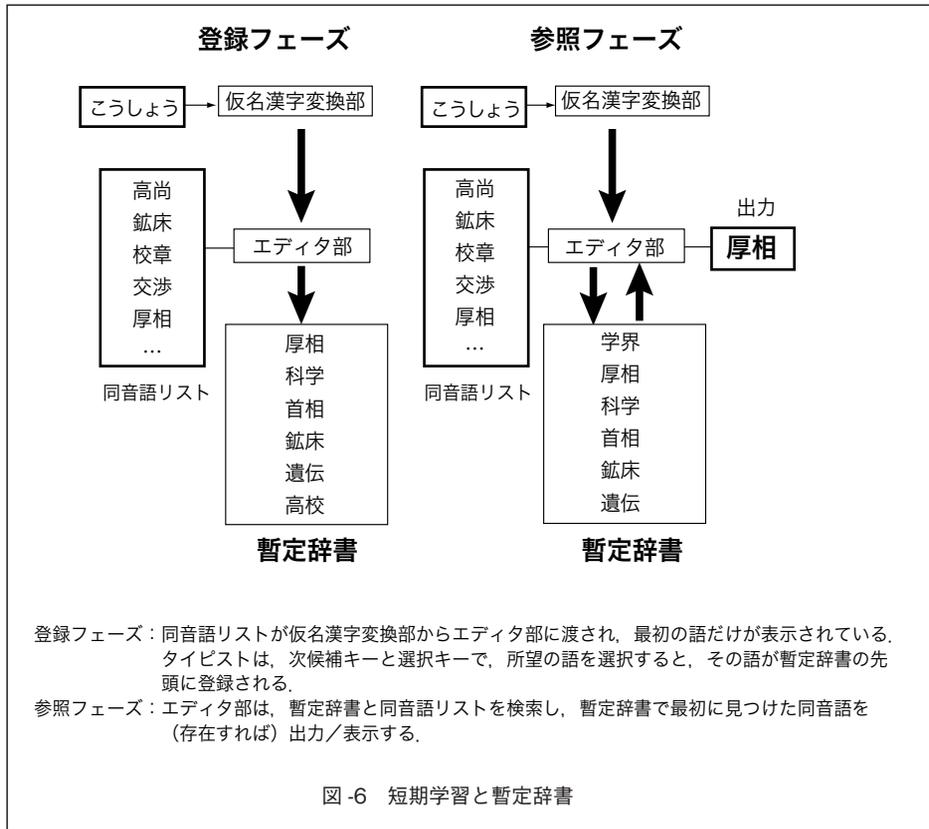
### エディタの方式

エディタ設計のための基本理念は、文書形式の柔軟性と、操作の統一性を確保することであった。文書は、A4で作ったものを、後にB4にしたいという場合がオフィスでは頻繁に発生する。WPは、文書を作成するだけでなく、文書の再利用によって事務効率を上げることを目的の1つにしている以上、従来のラインエディタのように行が操作単位になっていることは許されない。また、

<sup>☆18</sup> この数字は、シフトの影響を受けないテンキーから入力する。

<sup>☆19</sup> 「I」は固有名詞-in, 「J」は固有名詞-outのシフトを示す。

<sup>☆20</sup> 「住友」は固有名詞変換エンジン、「機械工業」は仮名漢字変換エンジンで変換される。



改行を行末までスペースで埋めることによって実現することも、削除された文字の後に空白を埋めることも許されない。このため、ディスク内の文書構造形式は、マークアップ方式を採用し、表示/印刷形式とは独立にした。筆者らはこの文書構造を当時、無構造と呼んでいた。このようなアイデアを規格化するためのSGML (Standard Generalized Markup Language) のドラフトが出てくるのは、ようやく1980年になってからである。モニタに文書を表示する場合、エディタはマークアップを解釈し、表示形式に展開し、さらに主記憶上の各文字がモニタのどの位置に表示されているかを示すキャラクタマップを主記憶内に作って編集校正に柔軟に対応した。文書を開くとき、モニタの下部に、「文書を展開しています」と表示されるのは、このことを示すものであった。図-7にJW-10の画面を示す。

■ヒューマンインタフェースへの配慮：

統一された操作性■

日本語WPは、これまで存在していなかった新しい概念の機械であった。さまざまなヒューマンインタフェース関連の困難が起きることは予想されていた。これに対処するため、基本機能はマニュアルレスで使えることを目指した。そのために、メニュー項目、編集キーの

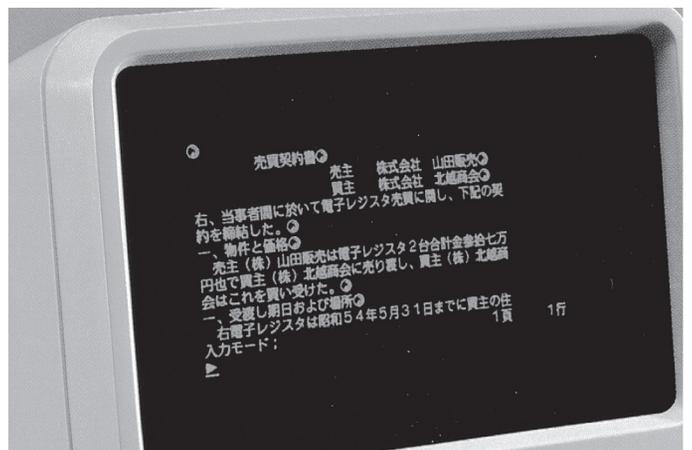


図-7 JW-10の画面

形状、名称、位置、そして操作の統一性には特別の配慮が払われている。基本操作は、「使いたい機能を示すキーで、校正したい文字をポイントする」というだけのものである。たとえば、ある範囲の文字を削除する場合、最初の文字にカーソルを移動して削除キーを押す、次に、最後の文字にカーソルを移動して削除キーを押すという操作であった。

## ワードプロセッサのハードウェア

JW-10のハードウェア緒元を表-2に示す。

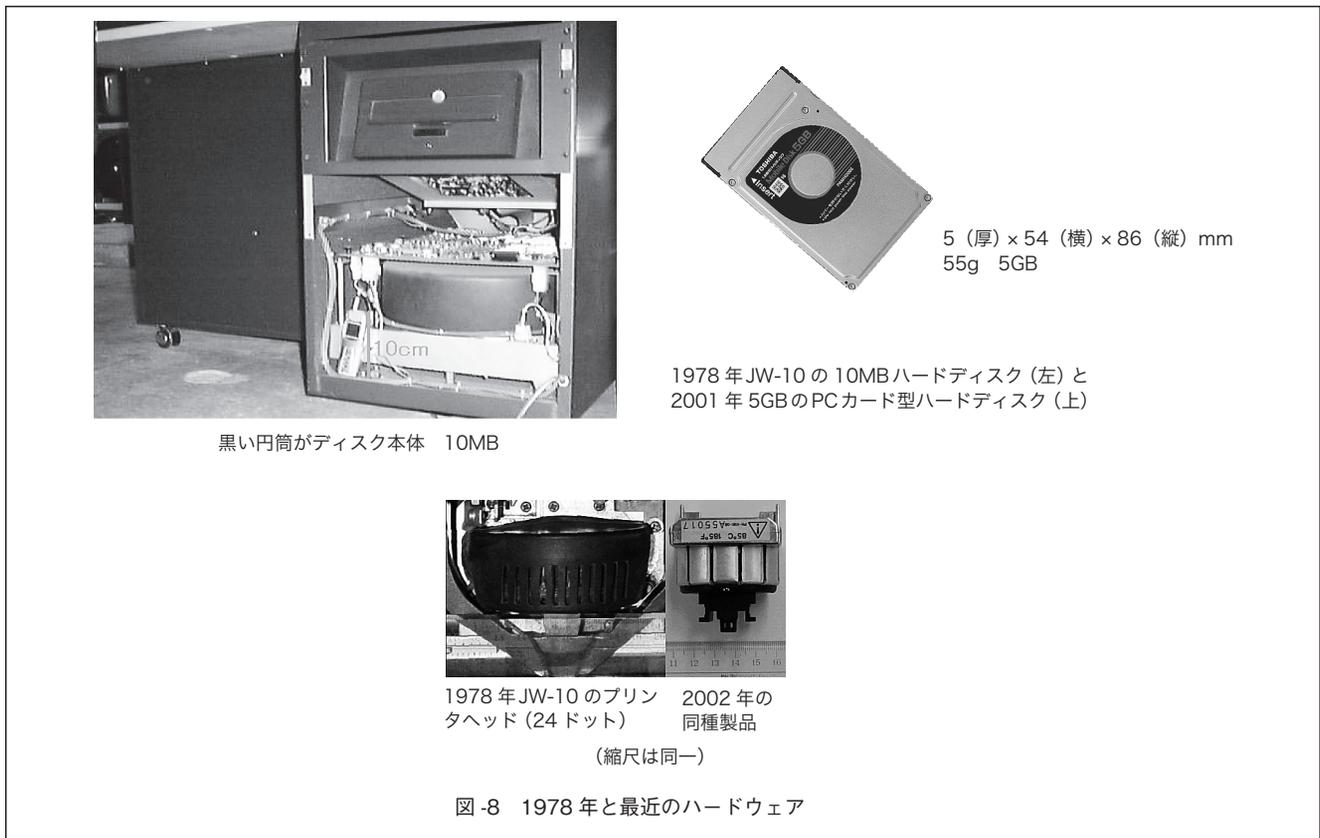
1978年当時、ハードウェアで最も問題であったのは、低価格で小型の漢字モニタ、漢字プリンタが存在していないことであった。これらは、JW-10のために新規開発された。漢字ハードウェアは当時、特機ともいうべきジャンルに属していたため、量産されず、非常に高価で

あり、また大型でもあった。JW-10が片袖機の大きさに収まったのは、これらハードウェアの新規開発によるものであったが、また、片袖機ほどに大きくなった主たる理由は、プログラム、辞書、文書を収める10MBのハードディスクのためであった。JW-10以降、JW-5のような8インチフロッピーだけのモデルも各社から発売され、小型化・低価格化されていったが速度の面では使いにくさは否めなかった。図-8に、当時のハードウェアと最近の類似ハードウェアの写真を示す。

CPU	約0.2MIPS
主記憶	64KB
ハードディスク	10MB
フロッピーディスク	250KB
キーボード	配列：JIS配列または50音順配列
モニタ	画面サイズ：12インチ、表示方式：24×24ドットマトリクス 表示文字数：448字(32字×14行)
プリンタ	印字方式：ワイヤードットインパクト方式 文字構成：24×24ドットマトリクス 文字サイズ：3.9mm×3.5mm(約11ポ) 印字速度：35字/秒 印字字数：最大90字/行 複写枚数：5枚(含オリジナル)☆ <sup>21</sup> 単票装置：オプション
設置緒元	寸法：(幅)1150mm×(高)1022mm×(奥行)760mm 電源：AC100V、単相800VA

表-2 JW-10のハードウェア緒元

☆<sup>21</sup> カーボンコピーの取れる枚数である。当時は、公文書などは法制上カーボンコピーが必要であるものがあり、重要な機能であった。





## ④ JW-10 以降のワードプロセッサの歩み

JW-10 以降のWPの歴史は、小型化、低価格化の歴史であるといつて過言ではない。1980年代半ばですでに、1行とはいえモニターは液晶化し、形態はポータブル化し、価格もJW-10の630万円から10万円を切るまでになった。ソフト面では、1980年代中頃には、べた入力、一文丸ごと変換、あるいは複文節入力と呼ばれる方式が各社のWPに実装されていった。複文節入力は、すでに1970年代末期から大阪大学などで研究されていたが、この時期になってようやく、実用化の環境が整ったのである。これは、漢字シフトキーも、文節キーも原理的には不要で、ただ仮名文字だけを入力していれば、WPが自動で文節を認識して変換していくものであり、より、英文タイプライタに近いものであるといえる。

一方、これと並行して、同音語の問題に対しても新たな取り組みが始まっていた。共起辞書による、1960年代のKatz and Fodorの意味論の実装である。これは、「暑いーお茶」、「厚いー夏」などの同音語問題を解決するのに非常に有効な方法であった。このような語あるいは、意味標識<sup>☆22</sup>の正しい組を収めた辞書を共起辞書と呼んで、ほぼすべてのWPはこの機能を実装していた。しかし、このような静的な選択制限では、「厚いー鉄板」と「熱いー鉄板」のような、正しさが文脈依存する同音語の問題を解決できなかった。これに対して筆者らは、語をノードとし、関係の強さをアークとするニューラルネットワーク構成し、文脈追従することによって動的に同音語を自動選択する方法を開発・実装し、ニューロワープロと呼んだ。

べた入力はどのWPにも実装されたが、この方式が有効に行われるためには、JW-10 開発時に問題であった、誤変換と同音語の問題に再度、新しい観点から取り組まなければならない。べた入力により新たに発生した文節の切れ目の曖昧性のために、文節の認識を誤ると、これまで以上に多くの誤変換を生産することになるからである。1995年、筆者らはこの事態に対応するため、品詞細分を一層推し進めた精緻化文法と呼ばれる機械文法を開発した。この新たな機械文法によって、形態素解析にとどまらず、部分的統語解析を実時間で行うことが可能になり、「ぜんぶかった」、「かんせんきのう」などの複文節、

あるいは複合名詞に対し、前部買った(正:全部買った)、幹線昨日(正:幹線機能)などの誤変換が生ずる問題を解決し、変換率の向上を果たした。

## ④ 文書作成機から家庭用汎用IT機器へ

こうして、WPは、急速かつ広汎に、オフィス、家庭に駆けこんでいったが、さらなる世界を開拓していった。家庭用の汎用IT機器としての世界である。1993年、モザイクによりWorld Wide WebブラウザがGUI(Graphical User Interface)化、1994年、カシオから低価格デジタルカメラQV-10が発売、1995年にはWindows95が発売され、マルチメディアとインターネットが大衆化していった。WPはこの動きを敏感にキャッチすると、それらの機能を取り込み、インターネットアプライアンス、マルチメディア機器としての家庭用汎用IT機器の役割をも担っていった。しかし、この道はパーソナルコンピュータと正面から競合することになる。その運用安定性を惜しまれながらも2000年以降、専用ワードプロセッサはほぼすべて姿を消し、現在、仮名漢字変換とエディタの技術は、パーソナルコンピュータ、携帯電話など、日本におけるあらゆるIT分野の漢字入力手段として引き継がれ、発展を続けている。

**謝辞** 日本語処理の研究に、ご助言、ご協力くださった元京都大学工学部長尾真教授、および、JW-10の研究開発に尽力された、元東芝総合研究所の河田勉、武田公人、元青梅工場の溝口哲也、児玉皓次、吉井清、および、WP関係の開発プロジェクトに参加されたすべての皆様に感謝いたします。

### 参考文献 (1967～1977)

- 1) 栗原、黒崎：仮名文の漢字混り文への変換について、九州大学工学集報、Vol.39, No.4, pp.659-664 (1967)。
- 2) 相沢、江原：計算機によるカナ漢字変換、NHK技術研究、Vol.25, No.5 pp.261-298 (1973)。
- 3) 松下、山崎、佐藤：漢字かな混り文変換システム、情報処理、Vol.15, No.1, pp.2-9 (Jan. 1974)。
- 4) 河田、天野、森：ミニコンピュータを用いたカナ漢字変換、電子通信学会技術研究報告 PRL76-47 (1976)。
- 5) 木村、遠藤、小橋：日本語入力用カナ漢字変換システムの試作、情報処理、Vol.17, No.11, pp.1009-1016 (1976)。
- 6) 天野、河田、武田：カナ漢字変換機能を備えたワードプロセッサ、電子通信学会情報部門全国大会講演論文集 (1977)。
- 7) 牧野、勝部、木澤：カナ漢字変換の一方、情報処理、Vol.18, No.7, pp.656-663 (July 1977)。

(平成14年10月3日受付)

<sup>☆22</sup> 単語を意味的に分類したグループの名称。たとえば、お茶、コーヒー、水、ワインなどは「液体」という意味標識を持つ。

