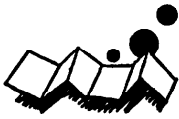


解説

2. 各界からみた DTP—ニーズの把握のために—



2.5 著者のための DTP の一実例†

石 畑 清††

1. はじめに

本を出版する際、従来は手書きの原稿を見ながら手作業で組版を行っていた。この作業は、人手によって1文字ずつ入力していくことになるので、日数、工賃ともかさみがちになる。これが原稿完成から出版までに時間を要する理由の一つになっていた。

最近では、組版を活字ではなく、計算機制御の写植機で行う例が増えている。これによって組版の手間がやや減ったと思われるが、人手を要する作業であることには変わりがない。写植機による組版の際は、原稿の文章そのもののほかに、印刷上の形式を整えるための制御コードの類を入力しなければならない。制御コードの量は、本の種類によって違うが、理科系の文書では一般にこれがかかなり多く、組版に要する労力が増える原因になっている。

一方、著者の側では、ワープロで原稿を作成する人が増えてきている。つまり、著者も印刷所も計算機で制御する機械を使っているわけである。したがって、人間の手を借りなくとも、ワープロから写植機へファイルを変換して移すだけで、組版作業の大半はすませることができるはずである。

この方法の利点として、第1にあげられるのは組版に要する時間や労力の節約である。また、植字の段階での字の誤りが追放できるので、校正の手間が減るという効果も見逃せない。

この方法で注意を要するのは、文字そのものは自動的に変換できるが、組版の制御に関する情報まで自動的に生成するのは無理だということである。ワープロ原稿には、本の印刷で使うような高精度の組版制御の情報が入っていない。したがって、変換の段階でワープロ中の情報から生成できる制御コードは限られたも

のだけになる。自動生成できる部分は変換プログラムで生成し、それ以外のは変換後の写植機ファイルに手作業で加えるという方針をとるのが得策である。もちろん、著者が印刷上の指定まで組み込んだワープロ原稿を作れば、完全自動化も可能だが、労力や品質の関係で適切な方法ではないと考える。

筆者は、この方法でこれまでに4冊の本を出版した^{†††}。ファイル変換のプログラムは自作したものを使った。本稿では、これらの本の出版の際の経験に基づいて、この方法による組版の詳細について述べる。

本の組版は、分野によって手間が大きく違う。数式を多用した文章は、最も困難なものの一つである。文字以外の制御コードが大量に必要だからである。逆に、日本語の文章だけであれば、制御コードはほとんど必要ない。筆者が出版した上記4冊の本は、いずれもプログラミング言語やプログラム作りの技術に関するものである。これらの本の特徴は、数式が少ないかわりにプログラムが多いことである。プログラムは、Pascal や Ada で記述されているので、予約語などを通常とは違ったフォントで印刷することになる。したがって、制御コードの量が比較的多い部類に属すると思われる。筆者の変換プログラムは、特にこの分野の本を能率的に組版することに重点を置いている。

ファイル変換による組版は、印刷を印刷所の機械で行う点、印刷上の細かな制御を著者ではなく印刷所の担当が行う点などで、本来の DTP という概念からははずれている。しかし、計算機を利用して組版を容易にするという目的や、著者が原稿作成の段階で印刷上の配慮の一部を分担することなどでは DTP と共通する点も持っている。他の DTP システムとこの方法の得失については、本稿の最後で述べる。

2. 印刷所における写植システムの概要

議論の前提として、印刷所で使っている電算写植システムの組版機能について簡単に説明する。機能の面

† Experiences of DTP for Authors by Kiyoshi ISHIHATA
(Department of Computer Science, Meiji University).

†† 明治大学理工学部

で、ワープロや DTP システムと本質的な違いはないと思われるが、印刷業界の習慣や品質を追及する必要から、考え方の違う点が散見される。

印刷所用の電算写植システムには、数社の製品がある。ここでは、このうち筆者が上記 4 冊の出版に使った 1 社のシステムについて述べるが、他社の製品も基本的な機能に大差はないと考える。

印刷用のシステムは、文字の種類が多い。JIS の漢字コード表にない文字も数多く含まれている。また、英字、数字、句読点、記号の類には、字の幅や字体の違うものが数種類ずつ用意されている。これらを文脈に応じて注意深く使い分けことが要求される。その他の漢字、仮名については、字そのものは 1 種類しかない。字の大きさや字体（明朝、ゴシックなど）は、制御コードによって選択する。文字のコード系は、通常の計算機とはまったく違ったものを使っている。

制御コードは、組版上のさまざまな指令を写植機に与えるためのものである。上述の字の選択のほか、字や行の間隔、特定の文字の印字位置の指定、段落や改ページの指示、その他いろいろな種類のものがある。柱（ページの上の隅に章や節のタイトルを入れること）やノンブル（ページ番号のこと）など出版の習慣に合わせたものも用意されている。

制御コードの多くは、モード切り替えの意味をもっている。たとえば、いったんゴシック体と指定すると、別の字体への切り替えの制御コードが現れるまで、ずっとゴシック体で印刷される。

制御コードで特に目につくのは、字の印刷位置を細かく指定するための機能が豊富だということである。いわば低レベルの印字位置制御が可能ようになっていく。字の半分の幅、1/4 の幅などの送りのほか、微細な位置決めもできる。また、ある文字を別の特定の文字の位置にそろえて印字する指定もある。

制御コードも、普通の文字と同じく、写植システムのキーボード上に機能ごとに一つずつキーが用意されている。テキストフォーマッタのようになんらかの文法をもったコマンドで指定するのは違って、むしろ専用ワープロに近い操作方法である。いったん入力したものを修正するときも、これらの制御コード一つ一つが画面上で見える状態で行う。WYSIWYG にはなっていない。

全般的な印象として、写植機による組版では、作業者が個々の場面に応じて細かな制御を行うことが要求されるように感じられる。もちろん、字詰め、字間調

整、禁則処理などの基本的な処理は自動的に行うようになっているが、細かな点までは計算機の自動処理が及んでいない。それぞれの文脈で作業者が低レベルの制御コードを数多く使って、その場に適した制御の方法を工夫しなければならない。このため、文書の種類によっては、制御コードを大量に入れなければならない。文字そのものの入力よりはるかに労力を要することがある。

3. ワープロファイルから写植機用ファイルへの変換

筆者が行ったワープロから写植機へのファイル変換の技法の細部について説明する。初めに述べたように、変換プログラムで生成できる制御コードは、なるべく自動生成することが要点である。これによる印刷所の労力軽減の効果は大きいと思われる。この作業を容易にするために、著者に原稿作成の段階での若干の協力を求める必要があるが、なるべくよけいな負担をかけることがないように配慮した。

3.1 文字の変換表

変換プログラムの最も重要な機能は、ワープロの文字を写植機の文字に変換することである。これは、基本的には 1 対 1 の変換であり、変換表さえ準備してあれば難しいことはない。筆者は、完全な変換表を作るのは手間がかかりすぎると考え、本を 1 冊作るたびにその本で使っている文字だけを順次変換表に追加していく方法をとった。現在でも、変換表に入っていない文字（主に漢字）は多いが、筆者が使う字の種類は限られているので、この方法でも特に不自由は感じない。現在、変換表に入っている文字は 1581 字、うち漢字が 1197 字である。また、最新の文献⁴⁾の出版の際に新たに変換表に加えた文字は、80 字程度である。

文字のうち、英字、数字、句読点類などは、変換表の作り方に工夫をする余地がある。たとえば、ワープロでは英字や数字に半角と全角の 2 種類がある。これらを写植機の違うフォントの文字に変換することが可能である。印刷上の区別を半角と全角の違いとして原稿の段階から入れておくわけである。しかし、普通は半角と全角を組織的に使い分ける習慣がないので、著者に不自然な心理的負担を与えることになりかねない。また、英数字のフォントを区別する必要もあまりないので、この方法は採用しなかった。

句読点を初めとする記号の使い方には若干の約束を設けておくこと後の作業が楽になる。横書きの印刷で句

読点に何をを使うかは、出版社の習慣によって違う（ピリオド（.）と句点（。）, コンマ（,）と読点（、）のそれぞれ選択）が、いずれにしても漢字1字分の幅で印刷される。一方、ピリオドやコンマは、別の目的でも使う。たとえば、節の番号を3.5などの形で表したり、小数点付きの数を書いたりする場合である。これらの場合は、1/2字幅や1/4字幅で印刷することが多い。

この二つの場合を区別するために、ワープロの中では句読点としてのピリオド、コンマを使わないことにする。句点、読点だけを使う。印刷でピリオド、コンマを使うとしても、それは変換の際に対処できる問題である。ワープロ中のピリオドやコンマは、半角、全角とも句読点以外の目的で使う場合に限ることにする。そして、これらを1/2字幅のフォントに変換する。このようにすれば、2種類のピリオドをワープロの段階から無理なく区別できる。この区別をしておかないと、変換プログラムの段階で違うフォントに変換することは不可能になり、印刷所における手作業による修正に頼ることになる。この程度の規約であれば著者に負担を与えることはないと考える。

記号の類は、ワープロにはそれぞれ1文字で入っているが、これを変換する際に、複数の文字やコードの組み合わせに展開しなければならないことがある。たとえば、引用符（“と”）は左右の対の形で使うのが普通である。そのため、字の中央より一方の側にずれた位置にくるようフォントがデザインされている。ところが、プログラムの中では、片方の引用符（”）を左右両側で使う。したがって、そのままでは左に片寄せた形になって見苦しい。これを避けるため、引用符は右に少しずらして印刷するように、前後に制御コードを入れる。これは、変換表に特別な扱いをするよう印をつけておいて、その印が現れたらプログラムで処理することにすればよい。

3.2 文字の幅と位置決め

ワープロでは、字の幅に半角と全角の2種類しかない。一方、写植機の字の幅は、フォントによってさまざまである。特に、英字は字ごとに幅が違う。しかし、空白を初めとする記号を除けば一般の文字について、印刷の際の幅を気にする必要はない。一方、記号類はフォントを変えればいろいろな幅のものを選ぶ。記号の変換法を決める際には、本の目的に適した幅のものを選ぶ必要がある。たとえば、日本語の文章中とプログラム中では、記号のフォントを変えたほうがよいことがある。特に括弧の類には注意を要する。

筆者はそこまでやらなかったが、ワープロの段階で半角と全角の括弧を使い分けて、違うフォントに変換するようなことも考えてよいだろう。

文献3)や4)で使ったセミコロン（;）のフォントは幅が狭いので、プログラムの中で使う際は、直前の字との間を1/4字分程度あけたほうがよいと考えた。文献3)では、これも変換表に特別な扱いをするよう指定する方法で対処した。しかし、文献4)では、ワープロの段階から1/4字幅の空白を明示しておく方針をとった。原稿の中で使わない特別な文字（文献4)の場合は半角の下線、_）をそのための記号とした。原稿の段階から、このような純粋に印刷上の指示を入れておくことは、著者の作業の妨げとなるので好ましくないが、この場合はやむをえないと考えた。セミコロンの直前に右括弧がある場合は間をあけないなどの細かな配慮が必要なので、機械的に処理するのが難しいからである。半角の下線は、ワープロ原稿を読む際はほとんど邪魔にならないし、原稿完成直前に一斉置換コマンドで入れればよいので作業の手間も少なくすむ。1/4字幅の空白は、ほかにいくつかの場合にも微調整の目的で利用した。なお、ワープロの全角の空白は漢字1字分の空白、半角の空白は1/2字幅の空白に変換するだけで十分である。英文中の空白は1/2字とは違う幅で印刷する習慣があるらしいが、1/2字幅でも問題はない。

プログラムのインデント（字下げ）は、PascalやAdaのプログラムの構造を印刷の形態で表す重要な情報である。これについては、ワープロに入っている原稿を単純に変換するだけで満足のいく印刷結果が得られた。ワープロでは、半角の空白をいくつか使って字下げを実現しているが、これをそのまま1/2字幅の空白に変換するだけで十分である。英字や数字の幅がワープロと写植機で違っているが、字下げについてこの影響を考える必要はないようである。なお、1/2字幅の空白がいくつも並んでいる場合は、できるだけ1字幅の空白に置き換えて、写植機上の作業が楽になるよう配慮した。

複数の行にまたがって行頭以外の特定の字の位置をそろえることは難しい。たとえば、

```
if x > 0
then writeln ('positive')
else writeln ('negative')
```

の場合、2行目の *writeln* と3行目の *writeln* はそろっていることが望ましい。ワープロならばこの二つ

をそろえることは簡単だが、写植機になると then の幅と else の幅が違うので、特別の制御コードを入れなければそろわなくなる。さいわい、Pascal や Ada のプログラムの清書でこのようなそろえ方が必要になる例は少ない。字そろえのほとんどが行頭の空白の数の制御だけで解決できる。したがって、このような箇所については手作業で修正すれば十分である。文献 4) では、ワープロ原稿の段階で字をそろえるべき箇所に特別の印を入れることにしたが、原稿が見にくくなるので感心できなかった。

ワープロの習慣として、半角の英数字と全角の漢字や仮名が並ぶ箇所では、間に半角の空白を入れることが多い。この空白は、写植機の上では有害である。写植機は、このような箇所に 1/4 字幅の空白を自動的に入れる機能をもっている。このため、変換プログラムは、原稿をいったん全部読み込んで、このような空白を取り除いた後で、変換を始めるようにした。ワープロの段階でこのような空白を入れないよう習慣づけておけば、この種の問題は生じないが、別の目的で用意した原稿を流用するような場合もあるので、変換プログラムとしては配慮しておくことが必要であろう。一般に、本原稿をワープロで作成する場合、ワープロ上の美観を整えるための工夫はなるべく避けることが望ましい。

3.3 フォントの切り替え

Pascal や Ada のプログラムは、いくつかのフォントを使い分けて印刷するのが普通である。Pascal の場合、begin などの予約語はボールド (太字) で、一般の識別子はイタリック (斜体) で印刷する。このためのフォント切り替えの制御コードを手作業で入れるのは手間がかかる。特に、上記 4 冊の本にはプログラムが数多く現れるので、編集者が原稿整理の段階で字体の指示を書き込むだけでも長時間の作業を必要とする。そこで、英字のフォント選択は自動化することにした。あらかじめ、ボールドなどの特別のフォントで印刷すべき単語 (予約語) をファイルに登録しておく。変換プログラムは、英字が連続しているかぎり続けて読み込んで、単語を構成し、その単語がファイルの中にあるかどうか調べる。あれば、ボールドに切り替えてからその単語を出力する。なければ、英字の標準のフォントを使う。Pascal の本の場合、標準のフォントはイタリックである。これで予約語と識別子のフォントの区別には完全に対処できる。この方法だと、普通の文の中に現れる英単語までイタリックになってしま

うが、それを普通のローマン (立体) に戻すのは簡単な作業 (イタリック体の制御コードを取り除くだけ) であり、数も少ないので問題ない。普通の文中でしか使わない単語 (たとえば、Pascal や Ada などの固有名詞) をローマン用のファイルに登録しておけば、さらに作業は楽になる。

3.4 変換の対象外にした部分

上記 4 冊の本では、わずかな数しか数式を使っていない。数式の部分は、変換で対処することは諦めて、印刷所の手作業で組版してもらうことにした。そもそも、ワープロの段階でも数式はうまく表現できないので、推敲可能な程度に読みさえすればよいと割り切って、プログラム中の数式のような形で入れてある。それを編集者が見て、数式の組み方を指示する方式をとった。

同様に、表も自動変換の対象とはしなかった。これも表専用の制御コードを生成するには、ソフトウェア開発のための作業量が増えて得策でない。もちろん、図も変換の対象外である。

4. 変換の効果の測定

文献 4) について、編集者が印刷所で修正するように指示した記録を見ると、修正の量が多いのは、章や節のタイトルの字体や字の大きさを変える箇所、プログラムの部分で行頭の位置や行の間隔を変える箇所、文章中の英字をローマンに戻す箇所、図や表のキャプションに関する箇所、数式を含む箇所などである。いずれも、指示の数はそれほど多くない。それ以外は、ほとんど変換結果そのまま最終的な組版が得られている。したがって、変換の際に制御コードを自動生成する方法が効果を発揮していると認められる。

筆者の変換の方法で組版の作業量がどの程度減ったかを客観的に評価するために、変換直後の写植機ファイル (印刷所に渡した時点のもの) と人手による修正を経た最終的なファイル (出版に実際に使ったもの) について、各種の文字や制御コードの個数を調べた (次頁の表参照)。対象としたのは、文献 4) の本文 (1 章～7 章、図や表は除く) である。

数字の見方についていくつか注意しておく。文献 4) の出版の際、印刷所に写植ファイルを渡した後も原稿自体にかなりの修正を加えた。したがって、修正前と修正後の各種文字の数字が一致するとはかぎらない。修正後の写植機用データは 12 個のファイルに入っているが、それぞれ先頭に約 300 バイトの制御コー

	修正前	修正後
文字		
ひらがな	114205	117021
カタカナ	19126	19686
漢字	68873	70821
英字	50326	50791
数字	6851	6621
記号	32779	34815
制御コード		
空白 1/1	10289	10341
空白 1/2	3785	3854
空白 1/4	6774	8247
段落	6365	5713
ローマン	12894	12983
イタリック	12212	11251
ボールド	2535	2991
その他制御コード	10722	13127
制御のパラメータ	31292	16277
合計	389028	384539

ドが入っている。これは、ファイル全体の印刷方式を与えるものである。この部分は、どのファイルにも同じものをコピーして入れてあるだけなので、このための作業の手間はわずかなものである。

フォント切り替えや空白を変換プログラムで生成した効果ははっきり現れているとみてよいであろう。制御コードの大半をフォント切り替えや空白で占めていることが分かる。

修正後のファイルを詳細に検討すると、制御コードの種類が大幅に増えている。これは細かな制御のためにさまざまな制御コードを駆使しなければならないことを示している。これらのコードまで自動的に生成するのは困難である。しかし、いずれの制御コードも頻度が少ないので、手作業で十分対処できる。

イタリックの切り替えの数が修正前から修正後にかけて減っている。この分がちょうど余分にイタリック指定をした箇所に相当する。その他、記号が増えたり、制御のパラメータが減っていたりするなど、変換に不十分な点があることを示唆すると思われるが、一般的には所期の効果をあげていると評価できよう。

5. 評 価

ファイル変換による出版と DTP など他の方法による出版との比較を行う。この方法の問題点や将来の見込みについても述べる。

本稿で述べたのと同様の手法でワープロから写植機へのファイル変換によって本を出版する試みは、すでにいくつか報告されている（たとえば文献 5)）。印刷所によっては、各種ワープロ用の変換ソフトを用意してある所も増えているようである。この手法で本を出版する例が今後急速に増えることは間違いない。

しかし、筆者が出版関係者から聞いた範囲では、この種の技法を導入したことによって、かえって苦労が増えた例が多いようである。本稿で述べたとおり、写植機ファイルには文字そのもののほかに多くの制御情報が含まれている。変換ソフトは、文字情報を変換するだけで、制御情報にはほとんど考慮を払っていないのが普通である。変換したファイルにフォント切り替えなどの制御情報を後から加えるのは、作業者が慣れていないせいもあって容易な作業ではない。初めから手作業で入力したほうが早かったという現象が生じても不思議ではない。

もちろん、この種の逆転現象は、事例が増えて担当者が慣れるに従って解消されるであろう。しかし、制御情報を手作業で入れなければならないという問題はいつまでもつきまとう。これに対して、本稿で述べたのは、制御情報のうちのいくつかをワープロファイルに含まれている情報から自動的に生成しようという試みである。著者の負担にならない程度の自然な約束を導入するだけで、制御情報の多くの部分を自動生成でき、組版の手間を大きく軽減できるというのが本稿の主張である。

この種の変換による組版がうまくいくかどうかは、文書の性格によって大きく左右される。最も簡単なのは、日本語の文章だけですむような文書である。文科系の本の多くはこの範疇に属するであろう。文字が正しく変換できれば、組版はほとんど終わったようなものである。理科系の文書になるとずっと難しくなる。

印刷所の組版では、同じ文書でも縦組みのほうが横組みより単価が安く設定されていると聞く。横組みのほうが組版の規則が複雑になっていることを反映しているであろう。たとえば、英語の単語が間に入るだけでも、単語の切り方や字の詰め方などに配慮を必要とする。特に、数式を多用した文書の組版はきわめて困難である。筆者の例のようにプログラムが多く現れる文書は、中間的な性格をもつと思われる。変換ソフトウェアに工夫をしておかないと、手作業が多くなって、自動変換の効果が十分に発揮されない。しかし、制御情報を生成する規則は比較的単純なので、自動変

換は十分可能である。これは、数式と違って、文字フォントの変更などの単純な手法で満足のいく組版が得られるからである。

TeX などの DTP ソフトウェアを使って著者自身が印刷用の版下まで作る動きが盛んである。これと比較して、ここで述べた方法の最大の利点は、著者の労力が軽減されることである。版下まで著者自身が作成するには、清書用ソフトウェアの機能をマスターする必要がある。文法を覚えるのはもちろん、さまざまなトラブルを解決するノウハウまで身に付けなければならない。これに比べれば、変換による方法の場合は、変換を容易にするための若干の規約に従う必要があるだけで、原稿作成時の心理的負担は無視できる程度である。

変換による方法では、印刷上の細かな苦勞は、出版社や印刷所の担当者に肩代わりしてもらうことになる。著者には、本の内容について考えなければならないことが多いので、この種の表面上の問題は、専門家に任せるほうが全体的な労力のバランスからみて勝ると思われる。むしろ、小規模な出版で、その種の専門家の助けを借りることが難しい場合は、著者自身でソフトウェアを使う方法によらざるをえないであろう。ここで述べた方法は、出版社による商業ベースの出版の場合でないと適用は難しいと思われる。

この方法では、著者の手元の機械としてはワープロだけがあればよい。高価なワークステーションや高精度プリンタを用意する必要はない。これも利点の一つにあげられよう。

印刷の品質について、DTP システムとの得失を論じることは難しい。印刷に使う機械そのものの性能も影響するし、組版の制御をどこまできめ細かく行うかにも左右される。ただいえることは、組版に関するノウハウを著者自身が考えるより、出版社や印刷所の専門家の知識を生かしたほうが得策だということである。専門家は、経験から得た多くの知識をもっている。これに比べて、著者はそれなりの見識をもっている。

いても、やはり素人である。筆者自身の経験でも、著者の考える組版の方針は一人よがりのものになりがちであると感じる。

ここで述べた方法では、数式を多く含むような文書を効率的に処理できない。これについては、TeX のようなソフトウェアが普及して、印刷所と著者の両者が共通のソフトウェアを使うようにならないかぎり、有効な手法はないと思われる。

6. おわりに

ワープロから写植機へ、文書ファイルを変換して移すことによって本の出版を能率的に進める一つの方法について述べた。手作業で原稿を組版する際に必要な文字情報の入力が必要になり、版面の体裁に関する制御コードの入力だけですむるので、組版に要する時間が短縮できる。特に、変換の際に細かな工夫をすることによって、制御コードの挿入を中心とする組版の労力を大幅に軽減できることを示した。

通常の DTP ソフトウェアを使う方法に比べて、著者の労力の軽減などの利点があるので、今後の出版の中心的な方法の一つになりうると考える。今後、本の内容の種類ごとに組版の規則を洗い出して、それを変換ソフトウェアに反映することによって、より効率的な組版が可能になるであろう。

参考文献

- 1) 情報処理振興事業協会編：最新 Ada 基準文法書，bit 別冊，共立出版（1984）。
- 2) 石畑 清，筧 捷彦，安村通見：Pascal の標準化—ISO 規格全訳とその解説，bit 別冊，共立出版（1984）。
- 3) 石畑 清，正田輝雄：Ada プログラミング，岩波書店（1986）。
- 4) 石畑 清：アルゴリズムとデータ構造，岩波書店（1989）。
- 5) 梅棹忠夫：ワープロにつながった印刷機，図書，pp. 22-26（1990-3）。

（平成 2 年 9 月 26 日受付）