

TR-83-012

中 文 自 動 排 版

計畫主持人：陳 克 健

參與人員：許永高・廖敏芳・陳宏渠・廖慶和。

計畫編號：NSC 72-0404-F001-01

執行期限：七十一年 10 月 17 日至七十二年 10 月 16 日

日 期：中華民國七十二年十一月一日

中研院資訊所圖書室



3 0330 03 00043 9

0043

書	考	參
借	外	不

摘 要：

中文自動排版，大致可分為兩種不同的形式，一種是固定形式，另一種是自由形式。固定形式的排版，系統根據使用者的排版命令，完成某一固定形態的排版，如書信、公文、表格等的排版。自由形式的排版在不違反排版規則下，自由安排版面使趨於最優秀最美觀的版面結果如報紙類。如果從處理的過程劃分排版工作可分為四個步驟：(1)小版面的編排(2)版面切割及組合(3)版面的修改訂正(4)版面的顯示。一個整合性的排版系統在於如何模組化各個步驟，以令自動、互動式或被動式三種方式完成一個通用性的規劃設計，本計畫的主要工作重點即為模組化的設計以達到各部子系統的完成及連接。

目 錄

第一章緒論	1
1-1 導言.....	1
1-2 模組化的排版	1
第二章文件表格處理	5
2-1 文件處理系統的架構	5
2-2 文件版面規畫系統	7
2-3 表格描述語言	9
2-4 內容編排子系統 (FDIS).....	32
2-5 FDIS 的功能	36
2-6 系統展望及結論	41
第三章報紙的排版	43
3-1 緒論	43
3-2 報紙排版的數學模式	44
3-3 啟發式的組版系統	49
3-4 結論	57
參考文獻	58

第一章 緒論

1-1 導言

近年來電腦的應用有個很戲劇化的發展，那就是某些領域中電腦對於處理算術、邏輯運算的能力反倒不如它處理非算術性問題的能力那般重要了。在這個領域中，電腦面對的不是繁雜的數字，而是文字（TEXT），包括字、片語、句子、名字、地址、公式……等等無奇不有，面對這些文字資料，電腦可以做許許多多的工作，而這些工作在傳統上則是要用人力花很多時間或者是根本無法解決的。這些工作包括文章字句的格式化編排（Text Formatting），語句的轉譯成不同語言之語句、語音合成……等等。而整個領域我們便稱為文字處理（Text Processing）。這個領域目前正在蓬勃的發展，雖然它能做的事，利用人力也可以完成，不像天文數字的計算那般的人力所不能及，但利用電腦來做總是那麼經濟快速，正確而完全。

中文資訊處理亦必為當前首要研究發展方向，因而中文的自動排版系統的發展刻不容緩。

1-2 模組化的排版

一個廣義的排版系統其功能應包括組版、改版及顯示。而應用的範疇有公文、報表、書信、報紙等。組版方式的不同可以歸納為下列二種形式。一種是固定形式的排版，另一種是自由形式的排版。所謂固定形式的排版常見於表格、公文、書信等這些資料的安排都有一個

固定的格式，因此排版系統只要根據使用者的命令把中文依命令安排段落位置等。而自由形式的排版則常見於報紙的排版，它是在不違反某些規則下，自由安排版面使趨於最優秀最美觀的版面。事實上這兩種形式看似不同，但是經過仔細的分析，我們可以把組版的過程分做兩部份。

(1)文字處理（稱之為排版）

在這種處理方式中，系統把整個版面視為一個固定的形式，很可能是一個方塊，而使用者要把文字（Text）填入其中時，必須在他的 Text 中夾雜著控制位置，排列的命令。例如何時換行，何時換頁，字體如何，字號如何等等。該處理系統便根據這些文字和命令做排版的工作。換句話說，版面要如何調整，如何編排均是靠使用者利用命令與文字內容來決定，故曰「與文字有關的處理方式」。

這裏我們提到的這些命令，我們常稱為 ROFF 命令，這種系統也可以稱為 ROFF 系統，常見的如 IBM 公司的 SCRIPT, PDP-11 UNIX 上的 NROFF。〔1〕

因此表格、書信、報紙的小樣等都是這一類的處理形式。但是如何定義出表格、文件格式、報紙的拼版，這些工作性質就和上述的文字處理不同。

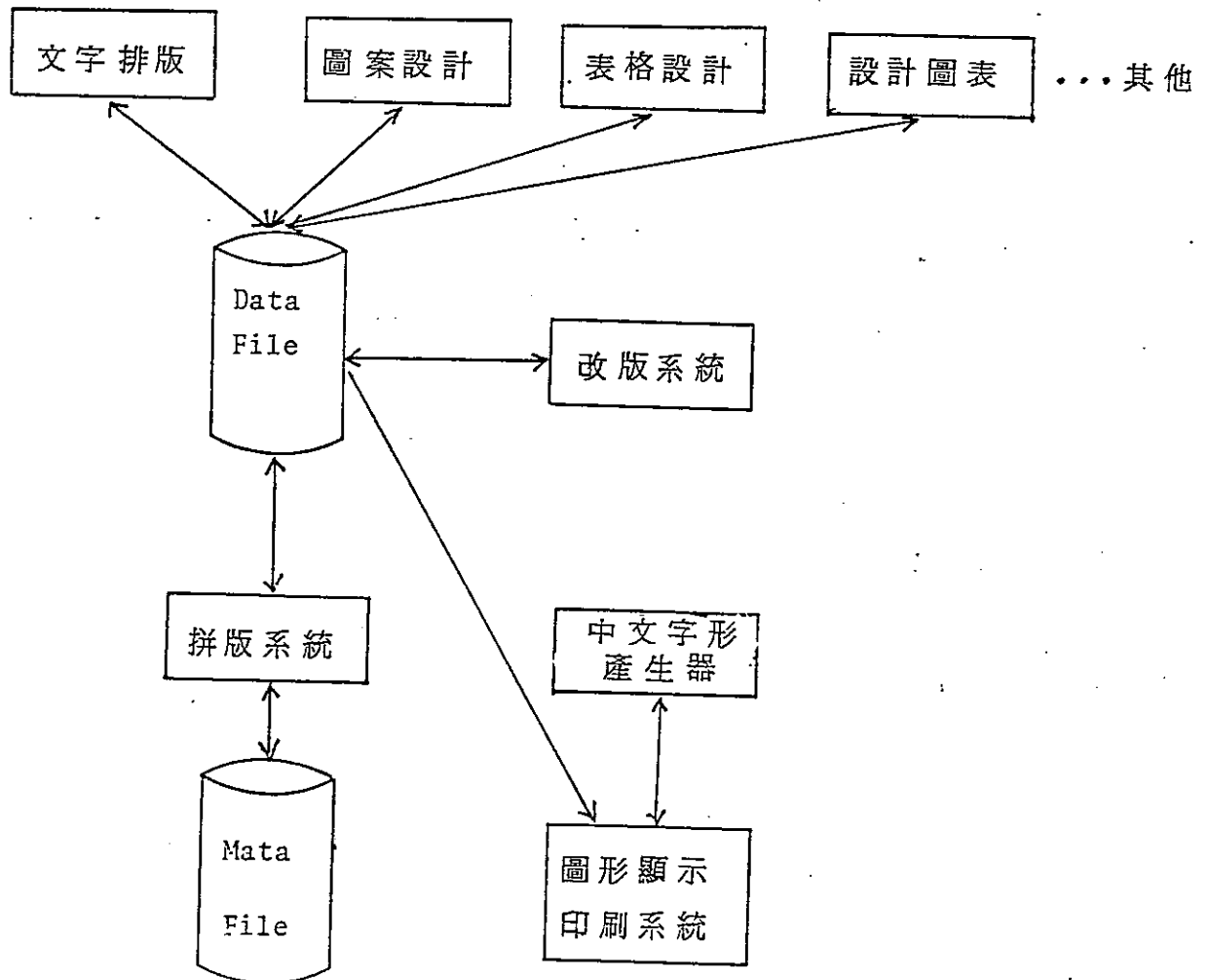
(2)版面分割或組合（稱之為組版）

我們把表格定義出來的過程，事實上就是把一個版面分割成一些小塊，例如從另一個方向來看就是把小塊組合成一個方形大版面。報紙的排版事實上就是把小樣合成一個大的長方形版面。信件、文件，

可視為一種最簡單的表格形式，也就是說它可能僅是一個方塊。像稅單、資產表格等，它們是由一些小方塊組成的東西。

從以上的分析可以了解一個整合形式的中文排版系統，可以模組化 (Modulized) 排版、組版、改版及顯示。各模組 (Module) 之間應該有共通的資料結構及中繼語言，(Meta language) 以利改版及顯示。

一個完整的電腦輔助排版系統可能規畫如下圖：



由於時間、經費及人力之不足，本計畫僅就文件表格處理，及報紙排版，完成了兩個單獨作業的子系統，系統之詳細說明及使用方法請參閱[2,3,4,5]。

第二章 文件表格處理

2-1 文件處理系統的架構

從使用者的觀點，整個文件處理系統可以分為下列四個部分〔參考圖 2-1〕：

1. 中英文字形產生子系統——負責字母的輸入 / 輸出處理功能。輸入時將鍵盤上的鍵入位置轉換為代表該字母的內碼。輸出時則根據內碼從中英文字型檔內找出對應於內碼的字母，並根據所要求的大小與位置將字顯示或印製在輸出裝置上。
2. 文件版面規劃子系統——提供一種描述語言和一些輔助工具，讓使用者得以方便地設計出所要的版面格式或表格框架。
3. 文件編排 / 填改子系統——提供一組方便的文字排版控制命令和一套交互式的填改工具，使用者可藉之在已定義好的文件框架內編排本文內容，或在內容格式已固定的表格上填入資料和修改已填好的表格文件。
4. 輸出驅動子系統——負責將具象文件顯示在螢幕上或繪製在指定的輸出裝置上。

從圖 2-1 中可以看出文件版面規劃子系統和文件內容編排 / 填改

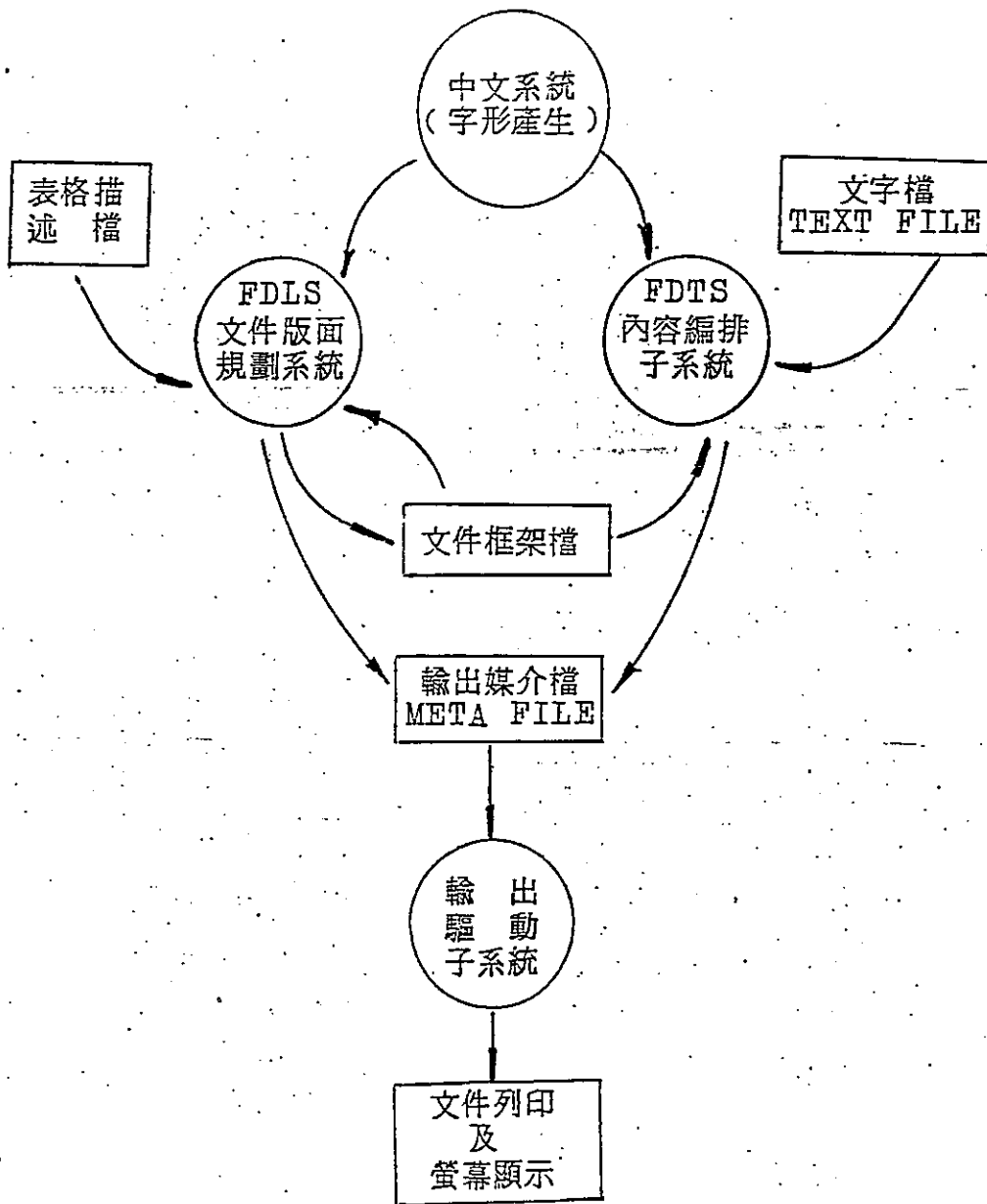


圖 2-1

子系統的介面是文件框架檔，兩者不直接交連而可以獨立操作。圖中的輸出媒介檔用來存放低層次的排版結果，使系統具有高度的裝置獨立性並合乎模組結構化設計的原則。

2-2 文件版面規畫系統

現有的文件排版系統都是將版面規畫和內容編排功能合在一起運作。這樣的系統將有如下的缺點：

- 文件準備者必須同時顧慮到版面的規劃與內容的編排。
- 辦公室中文件格式的設計者不一定是產生完整文件的人。這樣的系統就不允許一個人設計好的文件框架讓另外的人使用。

針對這些缺點而言我們認為把版面規劃功能和內容編排功能分離開來是有相當的用處。

此部分系統主要提供一種語法和語義都屬於高層次的描述語言。語言的細節與使用法將在下一節中介紹。此處列出有關語言的一些特性：

- 英文式的描述語句，使用者易於了解與記憶。
- 語言結構簡單，只有四種不同的語句形式。
- 語言中用以描述長度的單位是使用者所熟悉的米制單位（毫米）。
- 語言中採用相對坐標的觀念，使用者在設計版面時可集中注意力於較小的區域。

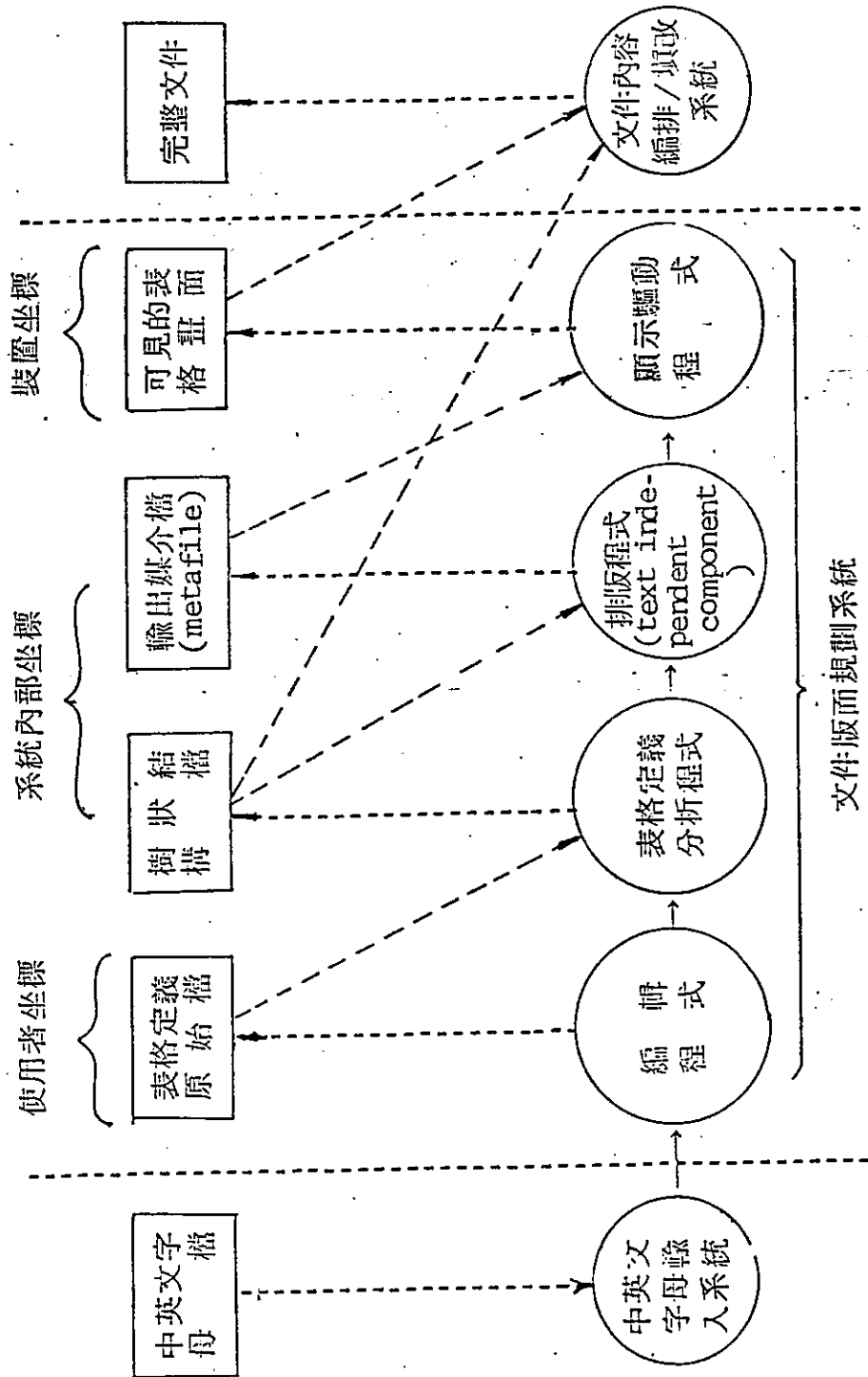


圖 2-3 文件版面規劃系統的處理流程

- 語言中採用由上而下的層次設計觀念，使用者可逐層設計表格結構，修改時也有層次性可循，不會產生牽一髮而動全身的現象。
- 此語言的製作乃採用多層次的坐標系統，因此具有高度的裝置獨立性。
- 此語言的製作是用高階語言 FORTRAN，再加上模組化的設計使系統具有高度的可攜性。

此部分系統的架構可歸述如圖 (2-3)。

使用本系統的程序大致如下：

- 步驟 1：使用者經由編輯器以 2-3 節所述的表格描述語言 (FDL, Form Description Language) 建立表格描述檔。
- 步驟 2：執行表格定義分析程式、排版程式和顯示驅動程式在繪圖終端機上得到表格畫面。
- 步驟 3：若對表格畫面有不滿意之處則回到步驟 1，利用編輯器修改表格描述，否則即可結束表格定義程序。

2-3 表格描述語言

本節將詳細說明一種針對表格框架之描述而設計的語言，我們稱之為 FDL (Form Description Language)。2-3-1 節將先介紹設計此語言的基本觀念，有了這種觀念後就不難了解語言本身。2-3-2 節接著說明 FDL 的語法和語義。

2-3-1 基本觀念

一般而言，語言的設計可以分為四個部分：觀念設計、語義設計、語法設計、和語彙設計。各部分都有其關心的主題，本節將說明 FDL 的觀念設計部分。

一般而言，設計表格框架時都是根據表格內容的邏輯意義將表格分割成幾個大區域如標題區、表目區、和註腳區等等，參考圖(2-4)。圖中的欄位分界線在目前是爲了說明才加上去，真正的表格描述中可以指定分界線的有無。

有了這幾個區域之後，我們才會繼續構想各區域的分割。例如 TITLE 區域可能要再分成四部分：機構名稱、單位名稱、表格名稱、和日期，如圖 2-5 所示。

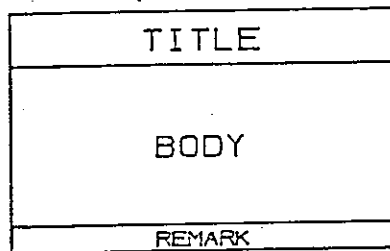


圖 2-4

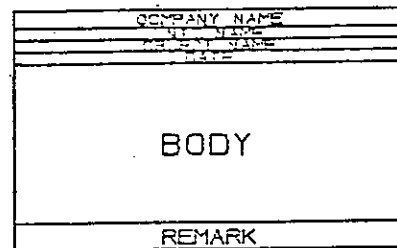


圖 2-5

同理其他欄位也可能逐層加細分割下去，先分割那一區域則無關緊要。此一過程若畫爲樹狀形式將可更容易看清楚，譬如上面所舉的例子可畫成如圖 2-6 的樹體。

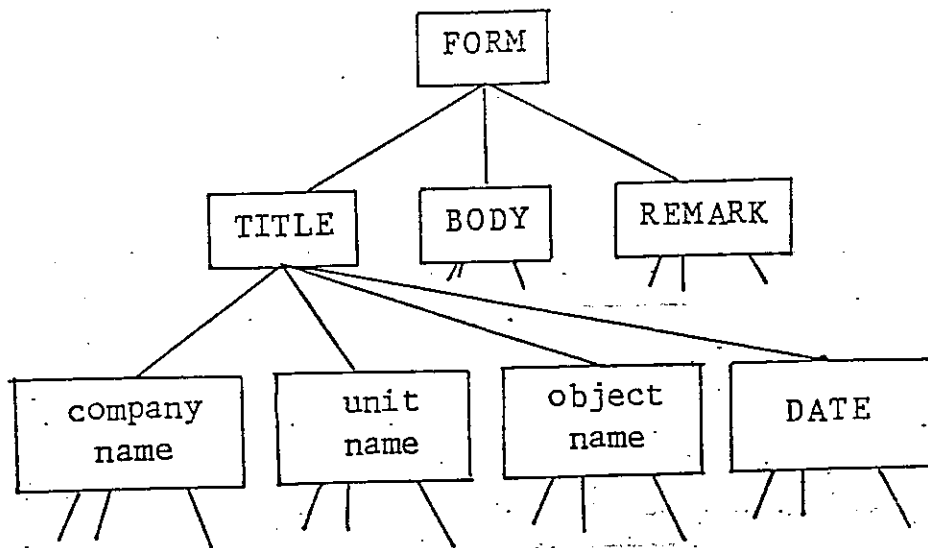


圖 2-7 表格的結構規劃過程

此種樹體狀分割步驟是 FDL 的主要觀念，有了它我們才可附加其他的觀念。

- 相對坐標與相對尺度的觀念 •

剛開始定義表格時當然先要決定表格的大小，此大小無可避免地要採用一種絕對的尺度單位。此後在分割過程中若是仍舊採用絕對單位和坐標，則各欄位的位置與大小將被牢牢地固定在表格坐標上。例如在 A4 紙張上設計上述的表格，假設 TITLE、BODY、和 REMARK 三欄位的縱向長度比例要為 4 : 12 : 2，若是採用絕對坐標，表格設計者就須先量度 A4 紙張（橫放）的縱向寬度，然後找出縱向坐標上 $4/18$ 和 $16/18$ 的坐標位置。這不僅運算困難而且很死板。因此使用者若能簡單地描述他要在垂直方向上分成三個區域，各部分的比例為

4 : 12 : 2 ，其餘絕對坐標的計算由系統來完成，將會使得設計者省卻相當的辛勞。此外更改畫面結構更是容易，我們以下面簡單的例子來說明，假設你已將表格橫向分割為五個部分，比例為 1 : 2 : 3 : 4 : 5 ，如圖2-7 所示。

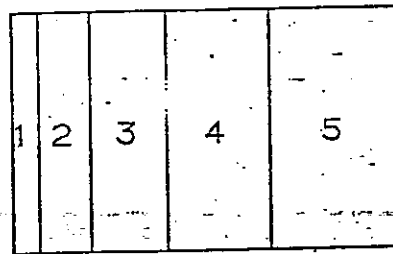


圖 2-7

若是要將各部分的比例改成 2 : 3 : 4 : 5 : 6 ，如圖2-8 所示。

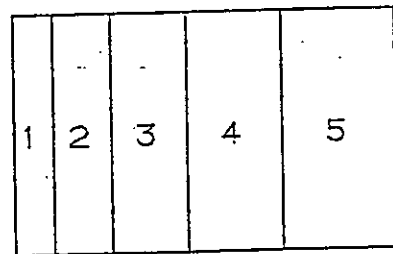


圖 2-8

或者要將欄位 1 和 5 的寬度互換，如圖 2-9 所示。

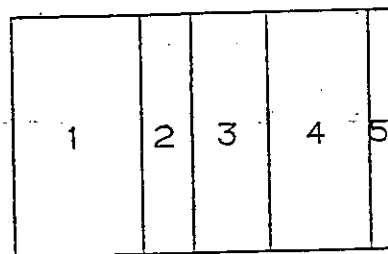
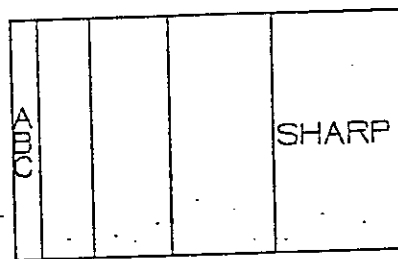
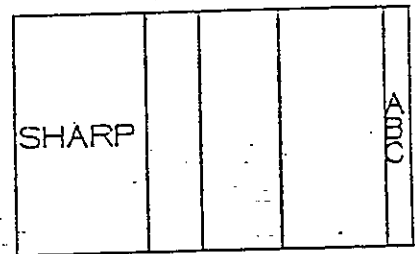


圖 2-9

你可以想像在絕對單位下等於要重新計算各分界線的所在，此外這些欄位的內容若已設計得很詳細，這種更動將帶給設計者無限的煩惱。此時各欄位的加細分割若採用相對坐標（亦即相對於目前要被分割的欄位），譬如前面的例子中分割 TITLE 欄位時以其左下角頂點為坐標原點，當父欄位（如 TITLE）的絕對位置有所更動時，將不會影響其內部的加細分割。若以上面的五部分分割之調整為例，設計者只要變動各欄位的相對位置，其內的物體即由系統自動調整絕對位置（參考圖 2-10），這樣將可避免設計者重新規劃各欄位的麻煩。



欄位 1 和欄位 5 對換之前



欄位 1 和欄位 5 對換之後

圖 2-10

• 屬性繼承的觀念 •

定義表格時，最後各欄位必須具有一定的屬性如用字的大小，字與字之間的距離，文字的對齊等等。在樹狀結構式的表格描述中，較低層次上的欄位乃包含在較高層次的欄位區域內。此時若採用遺傳的觀念（也就是祖先的特性可留傳成爲子孫的特性）使高層次欄位上所定義過的屬性自動保留在低層次的欄位上（除非低層次的欄位屬性重新被定義過）將可減少設計者許多重複描述的需要。譬如圖 2-11 的分割中。

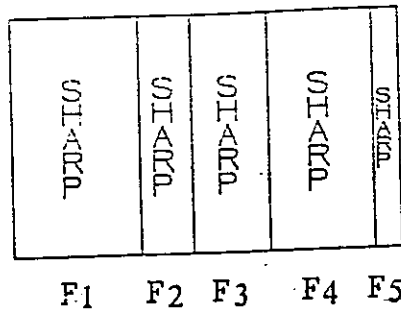
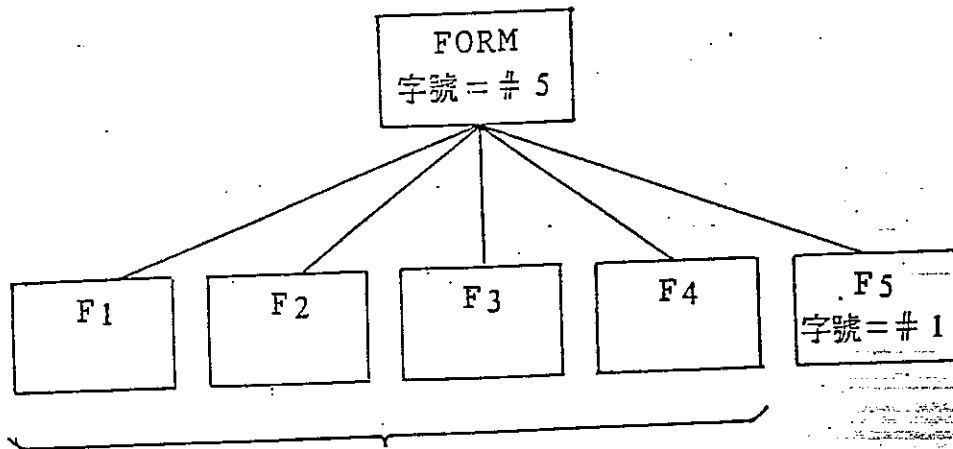


圖 2-11

欄位 F1, F2, F3, F4 都用同一號字體，欄位 F5 用另一號字體，則只要用圖2-12的定義即可做到，其中省卻了三次重複描述。



此四個欄位的字體大小將承繼 FORM 欄位的字體描述。

圖 2-12

此種屬性繼承的觀念和前輩所用“整體屬性”(global attribute)的觀念類似，但由於樹體結構的使用而更具彈性。當表格內容定義趨複雜時，這種屬性繼承的觀念將更形有利。

• 反復 (repeat) 的觀念 •

此觀念類似程式語言中的 DO 迴路。立即使用者可以利用反復描述之指示來促使系統實現同一類似的分割，例如在圖 (2-13) 中。

我們欲將畫面分為三部分，各部分內的分割型態都相同。此時若在定義過第一部分後，系統即能自動對第二部分、第三部分做同型態的分割，則表格設計者又可省卻不少的辛勞。在愈複雜的分割狀況下

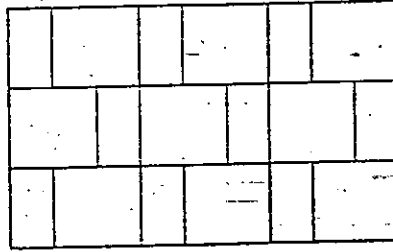
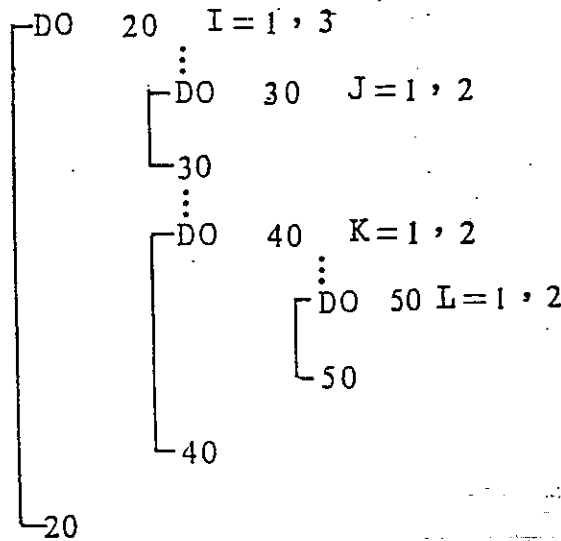


圖 2-13

，這種觀念的運用愈是有力。在修改時我們也只須修改有關第一部分的描述，第二、三部分即自動做同樣的修改。此一觀念和 DO 迴路同樣可以具有適當的包含關係，譬如和下列 DO 迴路：



相類似的版面分割樹體將如圖 2-14 所示。

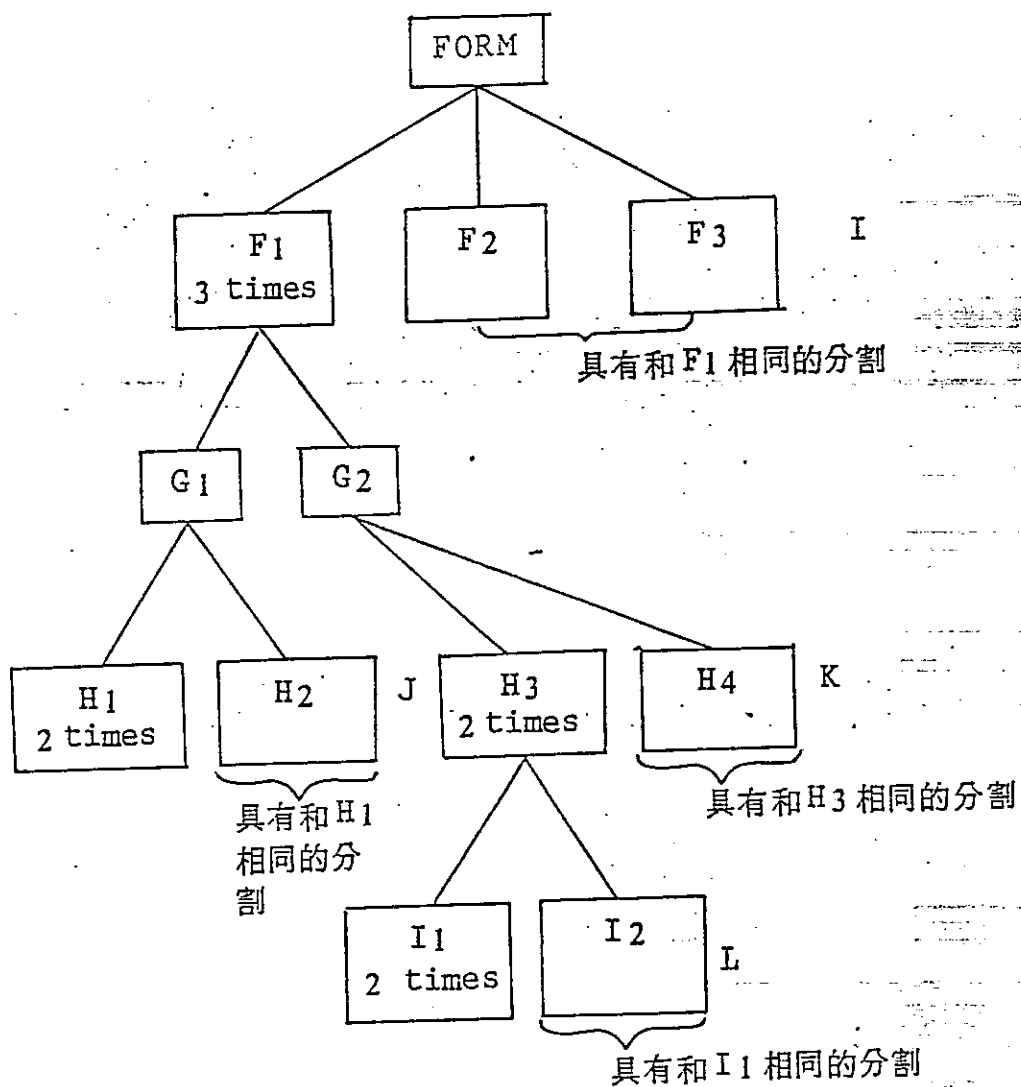


圖 2-14

2-3-2 語義及語法

就表格式文件而言，其版面描述語言應該具備的功能有如下幾點

1 選擇基本的具象物體：最常用的選擇是從指定的字模 (character font) 中取用字母和決定用字的大小。所謂字模即為一組字母所構成的集合，其中每一字母都具有同樣的大小和字型。其他在圖形中所用的具象物體有點、線、欄位大小等。在表格中常用的如欄位框架線，欄位間隔線，使用者必須能夠選定這些線的粗細。

2 決定物體的水平或垂直擺放：在文字方面，英文文章內很少有縱排的需要，處理中文則必須具備縱排的能力。其他物體如欄位等也有水平和垂直的相對擺置關係。

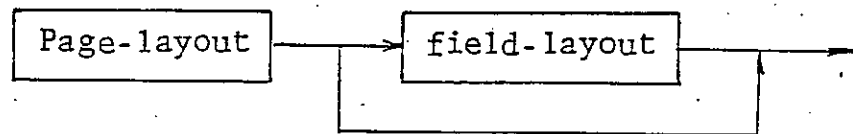
3 決定物體的放置位準：表格中常常有一些固定的欄位內容，設計者必須能指定其放置位準，例如靠欄位左沿或下沿對齊，或居中擺放等。

根據以上的功能需求，我們即可開始設計語言本身的語法和語義。

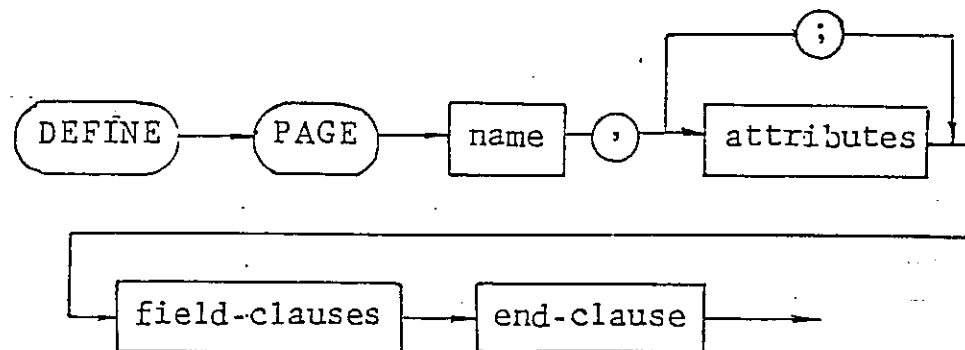
排版語言是一種具有特殊用途的語言。為了使大多數人（辦公室內工作者）都可以很快地學會自己規劃製作文件，這種語言必須具備很高的可讀性。使排版語言易於了解的一種方法是採用“宣稱式”（declarative type）的語法而儘量避開程序式的語句。由於文件的本質基本上是被動的，因而使用宣稱式的語言可說是一種合乎自然的選擇。利用這種形式的語言，使用者不需要寫作程式的觀念也很容易了解排版的過程。在宣稱式的語言之下，文件乃被視為一系列詳細描述文件架構與內容的宣稱。由抽象到具體的投映功能也是用一些附屬於文件的性質來加以控制。

下面我們用 18 個圖來表現 FDL 的語法。這種語法圖本身具有相當的說明能力，閱讀者很容易可以了解圖中所述的語法結構。此種圖也很容易轉換成 BNF 的形式，此處不再列出其對應的 BNF 描述。各圖中方塊內的小寫英文字是屬於描述語法用的字符 (non-terminal symbols)，其他大寫文字即 FDL 中的鍵語 (key word)，分隔符號，或特殊符號。

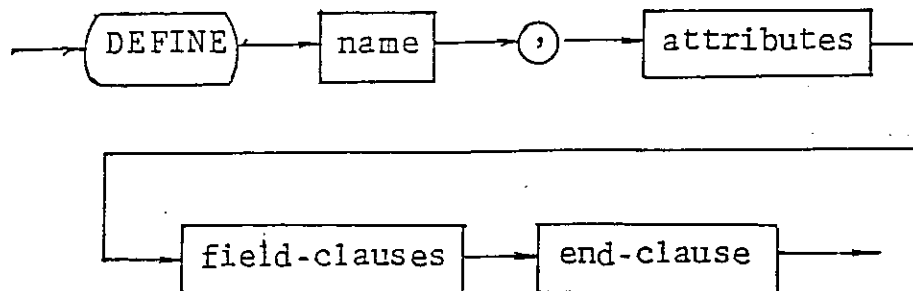
(R1) program :



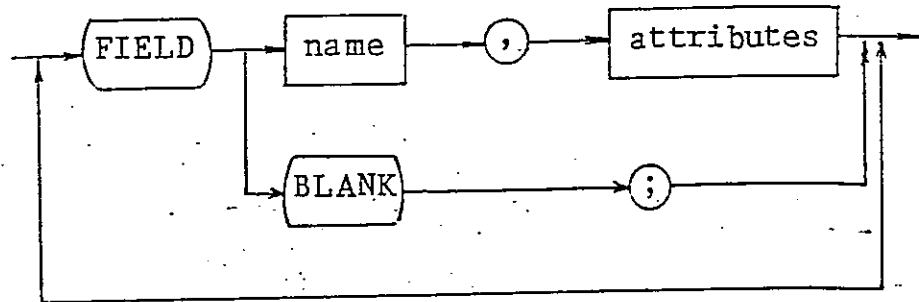
(R2) Page-layout :



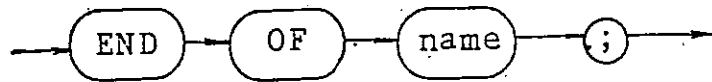
(R3) field-layout :



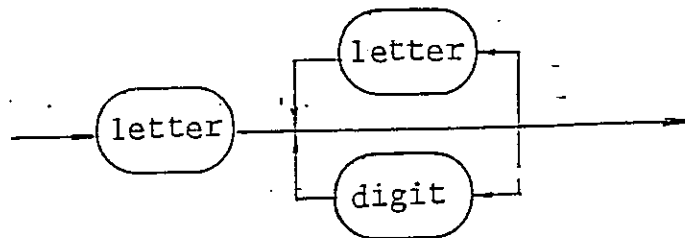
(R4) field-clauses :



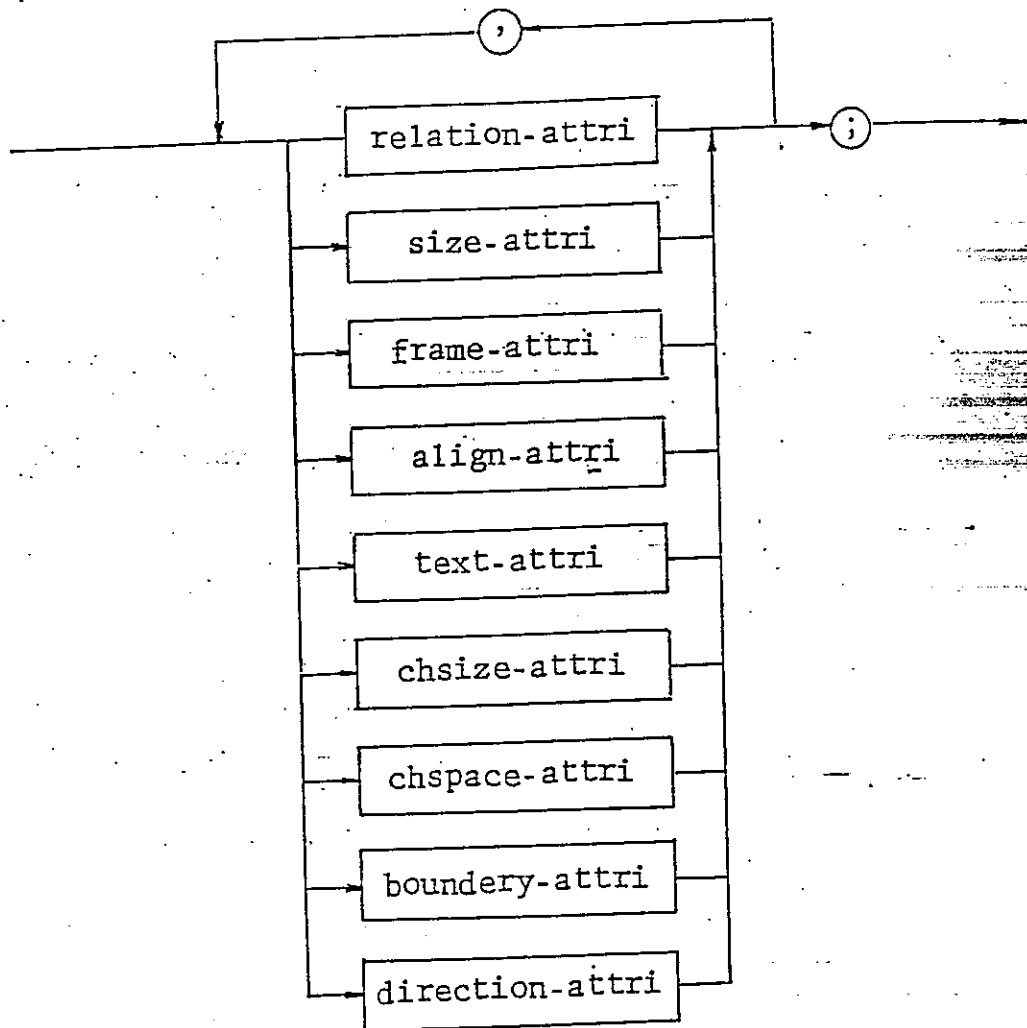
(R5) end-clause :



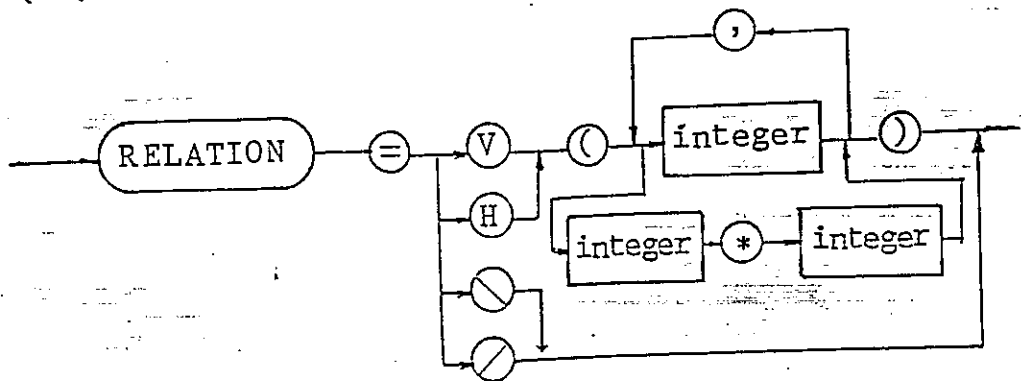
(R6) name :



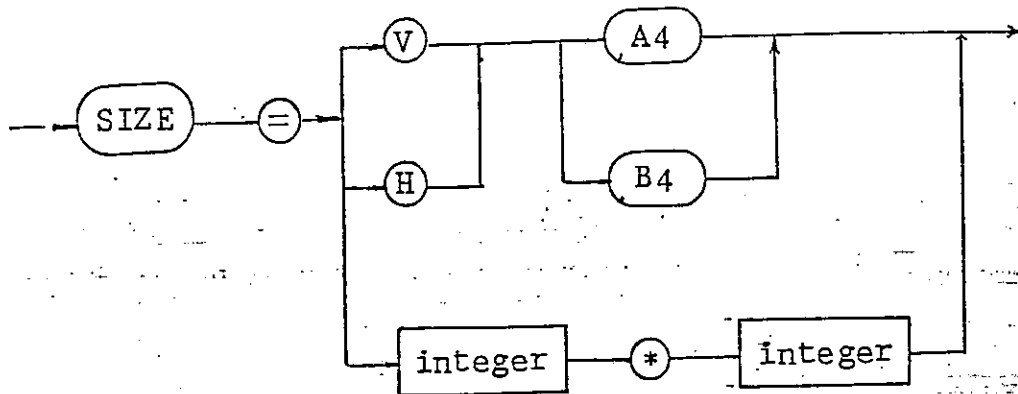
(R7) attributes :



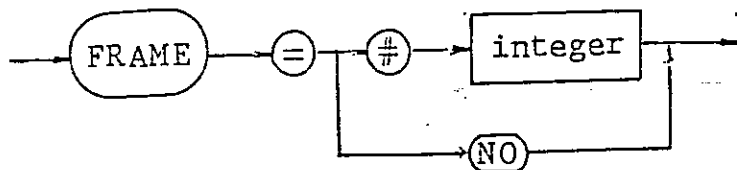
(R8) relation-attri :



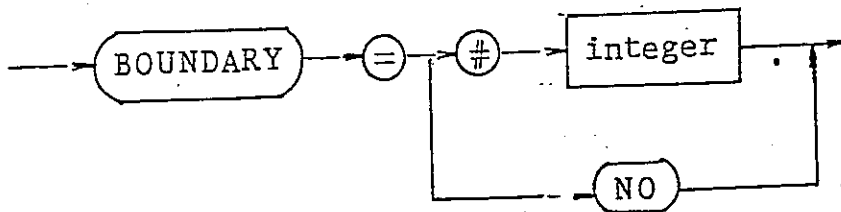
(R9) size-attrib :



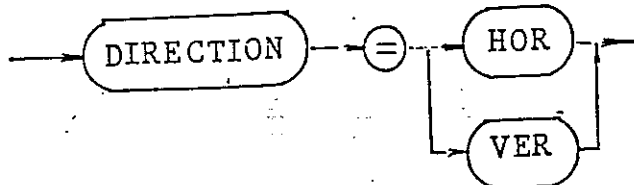
(R10) frame-attrib :



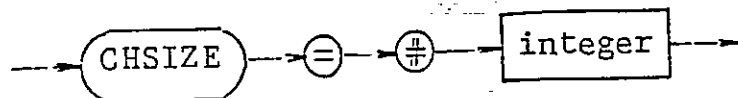
(R11) boundary-attrib :



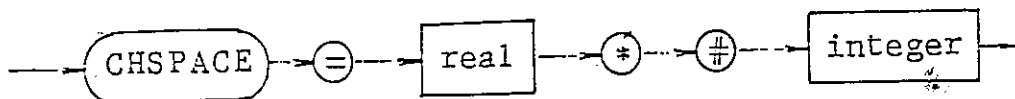
(R12) direction-attrib :



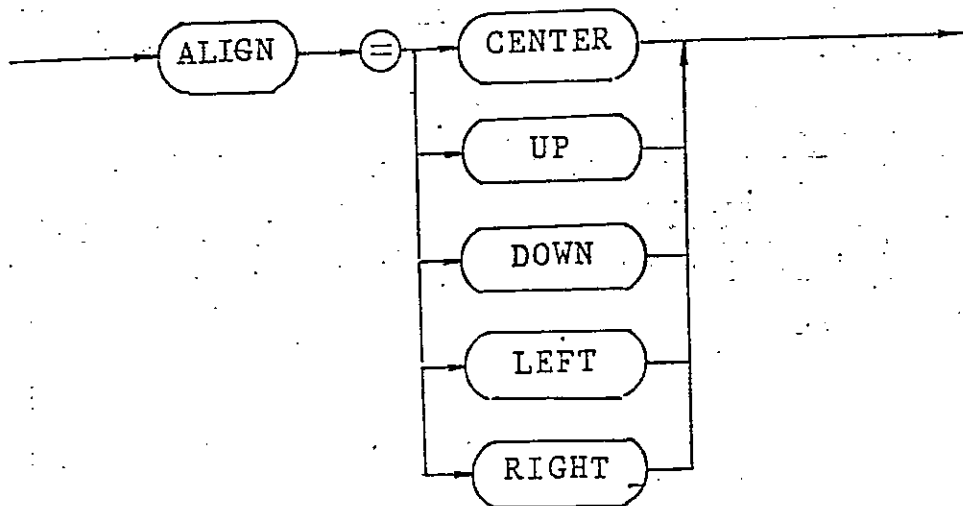
(R13) chsize-attrib :



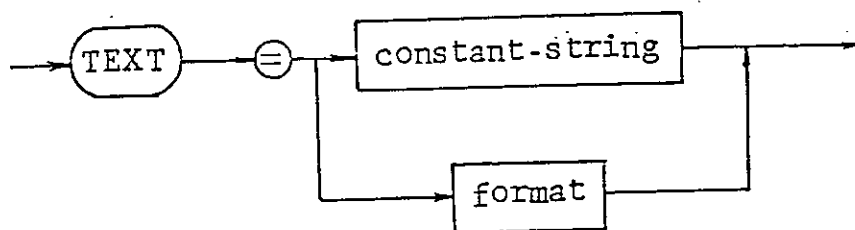
(R14) chspace-attri :



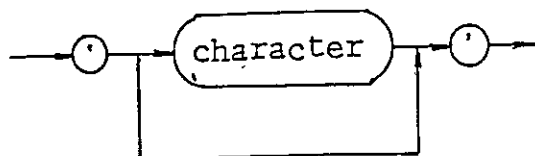
(R15) align-attri :



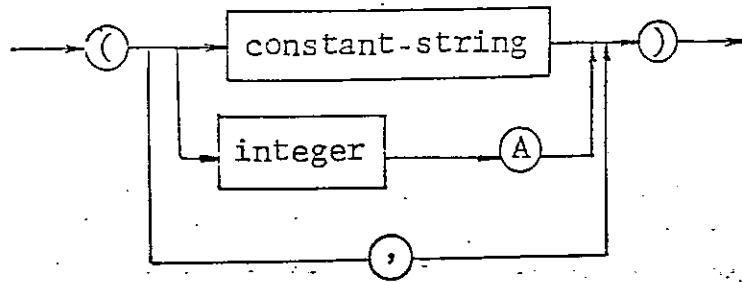
(R16) text-attri :



(R17) constant-string :



(R18) format :



說明：

(R1) : 整個表格的描述程式 (program) 分為兩部分 ; 第一部分是頁面的規劃描述 (page-layout) , 第二部分則為反覆規劃各欄位 (field-layout) 。

(R2) : 頁面描述以兩個鍵語 DEFINE PAGE 開始 , 然後設計者要給一個頁名 , 接着設計者可以定義一些頁面屬性 , 再接著定義各欄位 (field-clauses) , 最後用一個終結語句 (end-clause) 結束頁面的規劃。

(R3) : 欄位規劃和 (R2) 中頁面描述的語法相類似 , 此時少了一個鍵語 PAGE , 欄位屬性 (attributes) 的給定也和 (R2) 中所用有所不同。

(R4) : 欄位描述語句以鍵語 FIELD 開始 , 後面接著欄位名稱 , 再接著設計者可以擇定一些屬性參數。

(R5) : 此終結語句中的 name 要和對應的 (R2) 或 (R3) 語句中的 name 相同。

- (R6) : name 由英文文字 (letter) 和數字 (digit) 所構成，第一個字母必須為文字。FDL 所允許的長度為 8 個文數字。
- (R7) : FDL 中提供九種屬性片語，設計者可以按情況的需要指定各種屬性。
- (R8) : 分割關係屬性有四種選擇：垂直分割、水平分割和兩種斜向分割。前兩種分割型態以 V 和 H 代表，後面接著各成分的分割比例和反覆次數描述，各成分乃以逗號分隔來表示。
- (R9) : 頁面大小屬性分標準和非標準頁面，標準頁面以放置型態（橫放為 H，縱放為 V），後面接著標準頁面的代號來表示。非標準頁面則以“橫向長度 * 縱向長度”的形式來表示。
- (R10) : 框架線的指定以 # 符號後面接著一個整數線號來表示。用鍵語 NO 乃代表不需框架線。
- (R11) : 欄位分界線的指定和 (R10) 的語法相同。
- (R12) : 文字走向的指定有兩種選擇：HOR（水平走向）和 VER（垂直走向）。
- (R13) : 字母大小的指定以 # 符號後面接著字號來表示。
- (R14) : 字間距離以一個實數乘上字母大小來表示。
- (R15) : 欄位中字的擺放位準有四種選擇：居中放置 (CENTER)、向上沿 (UP)、下沿 (DOWN)、左沿 (LEFT)、或右沿 (RIGHT) 靠齊。

(R16) : 正文內容的給定有兩種型態：第一種為固定式 (constant-string)，第二種為固定正文與可變內容夾雜的形式 (format)。

(R17) : 固定正文乃以單引號來分離，正文部分不限字數。

(R18) : 可變內容部分以一個整數後面緊接著字母 A 來表示，意味著表格框架設計者預留那麼多個字位，將來表格使用者可填入或更改該部分的內容。

以上的語法和語義乃儘量保持其可讀性和表格設計應用指向的高

層次意義。下面為兩個實際的例子。

```
! THIS FILE DESCRIBE FINANCIAL FORM
DEFINE PAGE FINAN, SIZE=165*140, RELATION=V(10, 100, 5);
    FIELD TITLE;
    FIELD BODY, DIRECTION=HOR, CHSIZE=#1;
    FIELD REMARK, CHSIZE=#1, TEXT=( 'ABCDE NO : ', 10A);
    END OF FINAN;
DEFINE TITLE, RELATION=H(10, 55, 15);
    FIELD T1;
    FIELD T2;
    FIELD T3;
END OF TITLE;
DEFINE BODY, RELATION=H(10, 90, 5);
    FIELD B1, CHSIZE=#1, DIRECTION=VER, ALIGN=RIGHT,
        TEXT='ABC: ABCDEFCHIJK';
    FIELD TABLE, FRAME=#1;
    FIELD BLANK;
END OF BODY;
DEFINE TABLE, RELATION=V(10, 10, 10, 10, 5, 8, 10, 10, 10, 50),
    BOUNDARY=#1;
    FIELD TAB1, DIRECTION=HOR, CHSIZE=#1;
    FIELD TAB2;
    FIELD TAB3;
    FIELD TAB4;
    FIELD TAB5, ALIGN=LEFT, CHSIZE=#1, TEXT=( 'DATE F',
        2A, 'Y', 2A, 'M', 2A, 'D', 'TO', 2A, 'Y', 2A, 'M', 2A, 'D', 'E' )
    FIELD TAB6;
    FIELD TAB7;
    FIELD TAB8;
    FIELD TAB9;
    FIELD TAB10;
    FIELD TAB11;
```

```

END OF TABLE;
DEFINE TAB11, BOUNDARY=#1, RELATION=H(10, 20, 26, 12, 19);
    FIELD TAB111;
    FIELD TAB112;
    FIELD TAB113;
    FIELD TAB114;
    FIELD TAB115;
    END OF TAB11;
DEFINE TAB1, RELATION=H(10, 80), BOUNDARY=#1;
    FIELD TAB1C1, TEXT='NO', CHSIZE=#2, CHSPACE=1*#2;
    FIELD TAB1C2;
END OF TAB1;
DEFINE TAB1C2, RELATION=V(10, 10);
    FIELD TAB1C2U;
    FIELD TAB1C2D;
END OF TAB1C2;
    DEFINE TAB1C2U, RELATION=H(10, 20, 18, 18), ALIGN=DOWN;
    FIELD TAB1C2U1, TEXT='PPPR', CHSPACE=3*#1;
    FIELD TAB1C2U2, TEXT='FFFF', CHSPACE=1.5*#1;
    FIELD TAB1C2U3, TEXT='VVVVVV';
    FIELD TAB1C2U4, TEXT='EEEE', CHSPACE=1*#1;
END OF TAB1C2U;
DEFINE TAB1C2D, RELATION=H(10, 20, 18, 18);
    FIELD TAB1C2D1;
    FIELD TAB1C2D2;
    FIELD TAB1C2D3;
    FIELD TAB1C2D4;
END OF TAB1C2D;
DEFINE TAB2, RELATION=H(10, 80), BOUNDARY=#1;
    FIELD TAB2C1, TEXT='NM', CHSPACE=1*#2;
    FIELD NAME;
END OF TAB2;
DEFINE TAB3, RELATION=H(10, 80), BOUNDARY=#1;
    FIELD TAB3ADD, TEXT='AD', CHSPACE=1.2*#2, CHSIZE=#2;
    FIELD ADDRESS;
END OF TAB3;
DEFINE TAB4, RELATION=H(10, 30, 50), BOUNDARY=#1;
    FIELD TAB4C1, TEXT='BBBB';
    FIELD TAB4C2, TEXT=(BA, 'S', BA, 'SS'), DIRECTION=HOR,
        ALIGN=RIGHT;
    FIELD TAB4C3;
END OF TAB4;
DEFINE TAB4C3, RELATION=V(10, 10);
    FIELD TAB4C31;
    FIELD TAB4C32;
END OF TAB4C3;
DEFINE TAB4C31, RELATION=H(20, 10, 25);
    FIELD ASSNO, TEXT='XXXX', CHSPACE=1*#1;
    FIELD ASSCNT, TEXT='X X';
    FIELD ASSAREA, TEXT=('A A', '(SOME)');

```

```

END OF TAB4C31;
DEFINE TAB6, RELATION=H(18, 9);
    FIELD TAB6C1, TEXT='TE', CHSPACE=10*#1;
    FIELD TAB6C2;
END OF TAB6;
DEFINE TAB6C2, RELATION=V(10, 10);
    FIELD MON, TEXT='MON', CHSPACE=2*#1;
    FIELD UNIT;
END OF TAB6C2;
DEFINE UNIT, RELATION=H(1, 1, 1, 1, 1, 1, 1, 1, 1);
    FIELD U1, TEXT='M';
    FIELD U2, TEXT='T';
    FIELD U3, TEXT='H';
    FIELD U4, TEXT='T';
    FIELD U5, TEXT='M';
    FIELD U6, TEXT='T';
    FIELD U7, TEXT='H';
    FIELD U8, TEXT='T';
    FIELD U9, TEXT='D';
END OF UNIT;
DEFINE TAB7, RELATION=H(18, 9*1);
    FIELD TAB7C1, TEXT='H H H H H (D)';
    FIELD TAB7C1;
END OF TAB7;
DEFINE TAB8, RELATION=H(18, 9*1);
    FIELD TAB8C1, TEXT=('XX', 3A, 'XXXXXX', 4A, '%(DDDD)');
    FIELD TAB8C2;
END OF TAB8;
DEFINE TAB9, RELATION=H(18, 9*1);
    FIELD TAB9C1;
    FIELD TAB9C2;
END OF TAB9;
DEFINE TAB9C1, RELATION=H(15, 5), BOUNDARY=NO;
    FIELD TAB9C1L;
    FIELD TAB9C1R, TEXT='(HHHH)';
END OF TAB9C1;
DEFINE TAB9C1L, RELATION=V(1, 1);
    FIELD TAB9C1LU, TEXT=('RRR', 2A, 'Y', 2A, 'M', 2A, 'D');
    FIELD TAB9C1LD, TEXT=('BBBB', 5A, 'TTT');
END OF TAB9C1L;
DEFINE TAB10, RELATION=H(18, 9*1);
    FIELD TAB10L, TEXT='SS', CHSPACE=8*#1;
    FIELD TAB10R;
END OF TAB10;
DEFINE TAB111, RELATION=H(10, 10), DIRECTION=VER, BOUNDARY=NO;
    FIELD TAB111L, TEXT='OOOOOOOOO', CHSIZE=#2;
    FIELD TAB111R, TEXT=('KKK', 1A, 'LLLLL');

```

```

END OF TAB111;
DEFINE TAB112, RELATION=V(1, 1, 1, 1, 1), CHSIZE=#2;
    FIELD TAB112V1, TEXT='WWW', CHSPACE=#1;
    FIELD TAB112V2, TEXT='YYYY', CHSPACE=#1;
    FIELD TAB112V3, TEXT='OOOO';
    FIELD TAB112V4;
    FIELD TAB112V5, TEXT='TTTTTT';
END OF TAB112;
DEFINE TAB112V4, RELATION=H(1, 1);
    FIELD AA1;
    FIELD AA2, TEXT='D A';
END OF TAB112V4;
DEFINE AA1, RELATION=V(1, 1);
    FIELD AA1U, TEXT='R R';
    FIELD AA1D, TEXT='G G';
END OF AA1;
DEFINE TAB113, RELATION=V(1, 4*1);
    FIELD TAB113A, TEXT='W W(U)';
    FIELD TAB113B;
END OF TAB113;
DEFINE TAB114, RELATION=V(1, 5, 1, 5);
    FIELD TAB114A, TEXT='REDFSS';
    FIELD TAB114B;
    FIELD TAB114C, TEXT='TW', CHSPACE=2*#1;
    FIELD TAB114D;
END OF TAB114;

```

NO		PPPP	FFFF	VVVVVV	EEEE
NM					
AD					
BBBB		S	SS	XXXXX	AXXA (SQME)
DATE F Y M D T O Y M D E					
T		E		M O N	
H H H H H (D)				M T H T M T H T D	
XX XXXXXX % (DDDDDD)					
RRRR Y M D (HHHHH)					
BBBBB TTT					
S		S		REDESS	
W W W W W W (U)					
Y Y Y Y Y					
00000					
R R D A					
G G					
TTTTTTT					
XXXX					
TTTT					
TTTTTTT					
TTTTTTT					

ABCDE NO

Table 1. Results of the sorting process

! THIS FILE DESCRIBE ASSET CONTENT

DEFINE PAGE NO1, SIZE=HA4, FRAME=NO,
RELATION=V(10, 40, 120, 20);

FIELD BLANK;
FIELD TITLE, DIRECTION=HOR, ALIGN=CENTER;
FIELD BODY;
FIELD REMARK;

END OF NO1;

DEFINE TITLE, RELATION=V(20, 15, 15, 15);

FIELD T1, CHSIZE=#3,
CHSPACE=1.6*#3,
ALIGN=CENTER,
TEXT='ABCDE';

FIELD T2, CHSIZE=#3,
CHSPACE=.3*#3,
TEXT='ABAAAA';

FIELD T3, CHSIZE=#3,
ALIGN=CENTER,
CHSPACE=1.2*#3,
TEXT='ABAAA';

FIELD T4, CHSIZE=#1,
CHSPACE=1*#1,
ALIGN=CENTER,
TEXT=('AAAA', 2A, 'A', 2A, 'A', 2A, 'AAA', 2A, 'A',
2A, 'A', 2A, 'AA');

END OF TITLE;

DEFINE BODY, RELATION=H(10, 100, 10);

FIELD BLANK;
FIELD TABLE, CHSIZE=#6, FRAME=#2;
FIELD BLANK;

END OF BODY;

DEFINE TABLE, RELATION=V(10, 90, 10),
BOUNDARY=#1;

FIELD LABEL, CHSIZE=#1, DIRECTION=HOR;
FIELD CONTENT, CHSIZE=#1;
FIELD TOTAL;

END OF TABLE;

DEFINE LABEL, RELATION=H(20, 60, 10, 30, 20, 10, 40, 40, 60),
BOUNDARY=#2;

FIELD L1;
FIELD L2, TEXT='AAAA', CHSPACE=1.5*#1;
FIELD L3, TEXT='AB', DIRECTION=VER;
FIELD L4, TEXT='AB', CHSPACE=2.*#1;
FIELD L5;
FIELD L6, TEXT='CD', DIRECTION=VE
FIELD L7;
FIELD L8, TEXT='EF', CHSPACE=3.*#1;
FIELD L9, TEXT='CD', CHSPACE=3.*#1;

END OF LABEL;

```
DEFINE CONTENT, RELATION=V(20*1), BOUNDARY=#2;
  FIELD CONT;
END OF CONTENT;
DEFINE CONT, RELATION=H(4*5, 60, 10, 30, 20, 10, 40, 40, 60),
  BOUNDARY=#1;
  FIELD C1;
  FIELD C2;
  FIELD C3;
  FIELD C4;
  FIELD C5;
  FIELD C6;
  FIELD C7;
  FIELD C8;
  FIELD C9;
END OF CONT;
DEFINE TOTAL, RELATION=H(20, 170, 40, 60), BOUNDARY=#1;
  FIELD T01;
  FIELD T02, TEXT='A C ', ALIGN=LEFT;
  FIELD T03;
  FIELD T04;
END OF TOTAL;
DEFINE REMARK, RELATION=V(10, 20);
  FIELD R1, TEXT='ABCDEFGHIJKLMNPOQRSTUVWXYZ', CHSIZE=#1,
  ALIGN=UP;
  FIELD R2, CHSIZE=#4, CHSPACE=, 1*#4,
  TEXT=('8 C ', 6A, 'A A A ', 6A);
```

```

END OF REMARK;
DEFINE L1, RELATION=\\, BOUNDARY=#1;
    FIELD L1H;
    FIELD L1L;
END OF L1;
DEFINE C2, RELATION=H(3, 100), BOUNDARY=#1;
    FIELD BLANK;
    FIELD C2W;
END OF C2;

```

2-4 內容編排子系統 (FDTS)

FDTS 系統是由五個部份所組成的，其中除了 DEFVOL 部份之外，其他四個工作均要用到排版能力，因此我們稱這個排版部份為中心部份 (Kernel)。

簡單的說，Kernel 部份是一個 RUN OFF 系統 [1]，它提供了一些處理文字、片語的命令，和第二章述及的 PUB 系統的「低階命令」類似，使用者可以利用這些命令來填表或排版。也許要問，SCRIPT 系統中提供了「版面定義控制命令」[1]，為什麼在這裏看不到？事實上這些工作我們是在定義表格時早已定義完畢，因此 FDTS 不提供版面定義命令。

Kernel 所處理的檔案是一個循序檔案，檔案中夾雜著文字本身和命令，文字可以是中文、或英文，而輸出則是排好的結果，可在圖形終端機或繪圖機上看到。

2-4-1 Kernel 所提供的控制命令

Kernel 參考了 SCRIPT 系統，刪去了其中定義版面的命令部份，而為了能處理中文，又加入了一些功能而成，以下便是它所提供的命令敘述，圖 2-15 為一個 Kernel 的輸入資料範例，其中用括號包

合起來的部份即是控制命令，如 [NL]，[NP] 等等。

```
[CN2] [CS 1] [CN 03] [HHUUNNGG--CCHHYUU CCHHEENN., [NL] & [NL]
[CS 1] [SL 15] [ERD 10] [AADDVVIISOORR;: [NL] [CS 1] [CN 05] CCH
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[NL]XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
KKKKKCNF] LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA[NL]
```

圖 2-15

1 [V] 命令

這個命令允許使用者聲明目前排列的這個欄位的排列方式為直排 (Vertical)。通常它被安排在 [NF] (New Field) 命令之後。

2 [H] 命令

與 [V] 命令用法相同，但它是定義該欄位的排列方式為橫排 (Horizontal)。

3 [NL] 命令

這個命令告訴 FTDS 系統編排到目前的位置時，立即換到下一行繼續排列，[NL] 意為 New Line。

4 [SL] 命令

[SL] 意為 Skip Line，[SL .nn] 為其正確格式，其中 " b " 為 B Lank, nn 則是兩位數的整數，此外 [SLb01] 意義和 [NL] 命令是一樣的。

5. [SC] 命令

[SC] 意為 Skip Character，其正確格式為 [SCbn]，它的處理方式亦以字體大小為單位，若改變字體大小則 [SC] 命令的作用亦有相對的改變。

6. [NP] 命令

[NP] 意為 New Paragraph，FDTS 核心部份會把該命令分解為 [NL][SC02] 兩個子命令，各依定義執行下去。

7. [NF] 命令

[NF] 意為 New Field，顧名思義，它是告訴 FDTS 放下目前正在排列的欄位，直接跳到下一個欄位，繼續排列。在填表

8. [CS] 命令

[CS] 意為 Character Size，正確的格式是 [CSbn]，n 為 1~9 的整數，代表字號，使用者可以利用這個命令，隨時改變字體大小，核心部份對這個命令的處理過程如下：

(1) 改變排版時的 X, Y 指標。

(2) 把原來的字體大小放入堆疊中 (stack)。

(3) 改變字體大小。

其中第二步驟中的堆疊是一個 10 個元素的堆疊，它用來貯存字體大小，凡使用者聲明了一個新的字體大小時，舊的便放入堆疊中，直到使用者把目前的字體大小刪除之後，系統又回到原來的字體大小。

9. [DCS] 命令

[DCS] 意為 Disable Character Size，FDTS 遇到這個命令時便把目前的字體大小刪除，並到堆疊中取出最上層的資料作為新的字體大小。在處理 [CS] 以及 [DCS] 命令時，本系統要求使用者在該命令之前加上一個 [NL] 命令，亦即在同一行中，本系統不允許有兩種以上不同的字體大小，另外，[CSO] 意為清除堆疊中的所有資料。

10. [CN] 命令

[CN] 意為 Central Justification。完整的命令格式為 [CNbnn]，意為以下 nn 行做中央對齊的排列。若 nn 的數字大於整個欄位所容得下的行數時，處理方式和 [SL] 相同，即忽略去超出的部份。

11. [RD] 命令

[RD] 意為 Right-Down Justification，若該欄位為橫排，則為向右對齊，若為直排則為向下對齊，完整的格式為 [RDbnn]，nn 超過欄位的容量時，處理方式同 [CN] 命令。

12. [CN2] 命令

[CN2] 意為 Central Justification，但與第 10 個 [CN] 命令不同，[CN2] 是二維的中央對齊，當我們填表或排版時經常希望把某些文字放在整個欄位的中央，（幾何中心），這時除了聲明 [CNbnn] 之外便要再加上 [CN2] 來說明了。（圖

2-4-2 核心部份的處理過程

核心部份是一個類似 RUNOFF 的小系統我們稱之為 SET，它所能處理的單位是一個欄位，而它一次能處理的文字資料為 132 位元，在使用它來處理排版之前必須先設定一些參數，例如「欄位的長度」、「欄位寬度」，……等等。詳列於下，這些資料均是來自「樹狀結構檔」。

- 1 欄位長度
- 2 欄位寬度
- 3 欄位起始座標，X，Y
- 4 字體大小
- 5 橫排／直排
- 6 中央對齊與否
- 7 向右、向下對齊與否

根據以上各資料，再加上 132 位元組的「文字資料」（包括文字和命令），便可執行排版了。

2-5 FDTS 的功能

我們已經提過，FDTS 提供了五種功能方便使用者去處理表格，現在就這五個功能各別加以討論於後。

2-5-1 審視表格

當使用者透過 FDTS 選擇了這個功能之後，FDTS 便透過終端機詢問使用者三件事如下：

- 1 輸出裝置
- 2 樹狀結構檔案名稱
- 3 文字檔名稱

詢問完之後，FDTS 把控制權交給審視表格的副程式 (SEETAB)，於是 SEETAB 便執行以下各項工作。

- 1 根據樹狀結構檔的名稱去找到對應的表格媒介檔。(一個表格的樹狀結構檔和表格媒介檔的名稱相同，只有 Extension 不同。)
- 2 利用表格媒介檔把表格顯示到輸出裝置上。
- 3 找出表格文字檔並利用排版核心部份 SET 把各個欄位的資料編排入表格之中。

2-5-2 填寫表格 (FILL)

表格內容的處理中，最原始的動作便是填寫表格，當使用者要求執行填寫表格的功能時，系統也要問三件事：(1)輸出裝置；(2)樹狀結構檔名稱；(3)表格文字檔名稱。

接著 FDTS 便把控制權交給表格填寫副程式 (FILL)，該副程式便執行以下的動作。

- 1 找出與樹狀結構檔對應的表格媒介檔。
- 2 在輸出裝置上顯示出該表格。
- 3 建立一個新的表格文字檔，名稱由使用者定義。
- 4 利用 CROSS-HAIR 詢問使用者，使用者可利用它來告訴 FILL 該填那一個欄位。

5. 由文字終端機詢問使用者該欄位的內容，在本系統中，規定文字串的長度不得超過 132 位元組。
6. 讀取欲填寫欄位之描述資料。安排好 SET 程式的工作狀況。
7. 把 5 和 6 的結果交給 SET 程式執行。並產生文字媒介檔給輸出系統使用。

2-5-3 修改表格內容 (MODIFY)

在填表的過程中若發生錯誤，或者表格內容不當，要加以部份修改時，FDTS 系統提供了修改的功能，首先 FDTS 詢問使用者：(1)輸出裝置；(2)樹狀結構檔名稱；(3)表格文字檔名稱，詢問之後，控制權交給 MODIFY。MODIFY 便執行以下各個動作：

1. 找出表格媒介檔。
2. 顯示表格於輸出裝置上。
3. 找出表格文字檔，並填入表格中。
4. 顯示 CROSS-HAIR，讓使用者選擇欄位。
5. 接收使用者的修改命令。
6. 執行修改動作。
7. 把修改結果顯示於輸出裝置上。

2-5-4 DEFVOL 功能

DEFVOL (Define Volume) 之功能是在各種版面之上，加上一層定義，所謂「冊」(Volume) 就是一些版面的集合，而這些版面就是構成「書」的各種版面。

在「冊」的定義中包括了各個版面的名稱及其頁數，例如論文便是 $TCUT * 1 + DUAL * N$ ，使用者所定義過的「冊」我們存放在一個「VOL.DAT」檔案中，該檔案的結構如圖 2-16，其中「冊名」為使用者任意定義的名稱，而由第一種至第五種版面名稱則是目前已存在的五個樹狀結構檔的名稱，頁數為各種版面在排書時所重覆的次數。可由 $1 \sim 2^{16} - 1$ 。

DEFVOL 所使用的定義方式為「交談式定義」(Interactive)，如此使用者可免去記憶各種語法。定義的過程如圖 2-17：(a)，(b)，(c)。

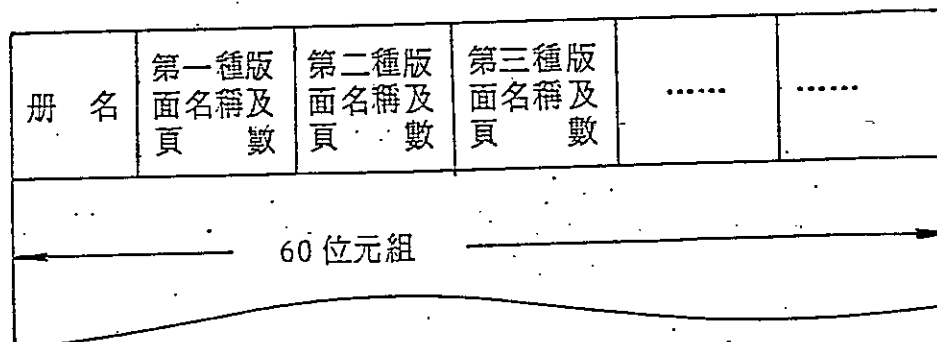


圖 2-16

PLEASE INPUT YOUR VOL NAME -- VOLUM1
PAGE NAME -- PAGE1
NUMBER OF PAGE -- 1
ANOTHER PAGE (Y/N) -- Y
PAGE NAME -- PAGE2
NUMBER OF PAGE -- 2
ANOTHER PAGE (Y/N) -- Y
PAGE NAME -- PAGE3
NUMBER OF PAGE -- 3
ANOTHER PAGE (Y/N) -- N
ANOTHER VOLUME DEFINITION (Y/N) -- Y

PLEASE INPUT YOUR VOL NAME -- VOLUM1
PAGE NAME -- PAGE4
NUMBER OF PAGE -- 4
ANOTHER PAGE (Y/N) -- Y
PAGE NAME -- PAGE5
NUMBER OF PAGE -- 5
ANOTHER PAGE (Y/N) -- Y
PAGE NAME -- PAGE6
NUMBER OF PAGE -- 6
VOLUME NAME ALREADY IN USED
RENAME OR COVER (R/C) -- R
NEW VOLUME NAME -- VOLUM2
ANOTHER VOLUME DEFINITION (Y/N) -- Y

PLEASE INPUT YOUR VOL NAME -- VOLUM1
PAGE NAME -- PAGE1
NUMBER OF PAGE -- 1
VOLUME NAME ALREADY IN USED
RENAME OR COVER (R/C) -- C
ANOTHER VOLUME DEFINITION (Y/N) -- N

2-17 (a)(b)(c)

2-6 系統展望及結論

FDTS 至目前為止，只是一個實驗階段的系統，許多功能都待加強發展，雖然在設計之初，我們花了很多時間去構想它應具備的功能，然而實作之過程中我們又陸陸續續發現一些可以加入 FDTS 或 FDLS 的功能，這些功能有些是可以直接想出如何處理的，而有些則只是一個概念，還須要多方面的設計，仔細研究才能解決的。

2-6-1 系統可增加的功能

這一節中，我們介紹一些可以加入 FDTS 或 FDLS 系統的功能，這些功能均是比較具體、容易想像的。

1 局部放大的功能 (Partial Scaling)

由於輸出裝置的「解析度」(Resolution) 有其極限，因此較細小的表格在螢幕上也許不易看清楚，因此若加入局部放大的功能，必可提高系統可用性。

2 欄位之間的運算 (Inter-field Operation)

經常在表格中，可看到「合計」欄，若在 FDL 中直接定義該欄是由某些欄位所合計而成，並經由 FDLS 轉換成某種內在形式，如此 FDTS 便可自動產生「合計」欄的內容。

3 一致性、完整性的檢查 (Integrity)

表格中，某些欄位的內容可能具有其一定的範圍，例如「性別」則只可能是「男」或「女」，而「百分比」則只可能介於 1 ~

100 之間，若在 FDL 中能加上這些範圍定義，則 FDTS 便可幫助使用者在填表時避免一些可能的錯誤。

4. 文件的機密性 (Security)

對某些使用者，也許有些文件應列為機密，若我們在 FDTS 中也提供一套「機密性」的檢查系統，則可順利達成這個目的。

5. 檔案系統 (File System)

目前 FDTS 對於檔案均未加管理，使用者必須自行注意某某檔案是做什麼用途的？某一個文字檔相對的樹狀結構檔又為何？這些細碎而大量的問題應有一個檔案系統來管理。

以上只是略舉數端，其他尚有許多可加入的功能，有待將來繼續發展。

2-6-2 FDTS 的遠程目標

FDTS 的遠程目標是一個智慧型的排版系統，除了具備 2-6-1 節中的功能之外，還希望能與資料庫系統介面。這是一個很粗略的概念，至於如何介面，將是一個研究的方向，原則上，希望 FDTS 在填表時，各個欄位的內容均可由資料庫系統中取出，亦即填表時直接用查詢語言 (Query Language) 來取出資料並填入。

FDTS 和資料庫介面之後會有什麼新的功能發生呢？這是一個值得想像的問題。

第三章 報紙的排版

3-1 緒 論

如何使報紙截稿時間延後，出版時間提早，為當前報業界力圖解決的問題，當然電腦化的作業為必然的趨向，電腦化自動排版有下列的問題必須一一加以解決的。這些問題有些已經解決，有些尚待研究，有些解決了但不甚很滿意。在這裏不詳述這些問題的解決方式，不過提供了這方面的參考資料。

- 1 新聞稿的輸入問題
- 2 中文字體、字形的產生問題
- 3 小樣編排問題
- 4 自動組版問題

新聞稿的輸入電腦作業，中文輸入鍵盤解決了大部份的問題 [6,7]，其餘照片圖片的輸入輸出仍有困難，所以不採用直接輸入圖形的方式，只輸入圖片的大小以利自動編版的作業。中文字體字形產生問題，也是一個非常嚴重而且難以解決的問題 [8]。並不是此一問題不可解，而是如何以較低廉的代價完成不同大小不同字體的大量中文字。因為如果硬把數萬個中文字以不同大小不同字體的影像存放電腦的記憶體內，需要的記憶空間約要上千個百萬拜 (mega bytes)，而這些字體的影像 (bit map)，又應如何產生呢 [8]？

至於自動編排版面的工作，可以分成兩個獨立的作業。先做小樣的編排，然後完成整版的組合作業。譬如說社論、新聞稿、氣象等獨立的新聞個體的內容，如何安排成固定形式的版樣即為小樣的編排。此一部份的研究較多，也到了實用階段〔4,9,10〕。把小樣如何填入大的版面就是自動組版的問題〔11〕。

組版是編輯工作中最具藝術性的一種，怎樣從版面上的美來吸引讀者注意，使感覺舒適，印象深刻，要從版面的組合及標題字體的選擇來達成。如何把這種組版藝術納入電腦中，並非一件容易的事情。在這裏僅就可以用數學模式表示出的因素，以評分的方式估計出版面的優劣。而有一些不易加以簡單模式評估的部份，仍由人為的編輯決定。例如新聞稿的選取，標題的大小及字體的選取，方塊形式或非方塊形式的決定等，這些是很難用一個測度 (measure) 來決定其美醜。所以目前測度版面評分因素，只能限於可測度的資料，以扣分方式給予版面一個分數，譬如版面的空白行，標題重疊，太多的小塊等皆應扣分。因此如何訂出測度及計量的方式為以數學模式探求較佳版面的主要工具。

3-2 報紙排版的數學模式

Knuth 在英文排版上定義了美觀的測度數學模式〔12〕，然英文排版所考慮的因素和中文排版有所不同，英文排版的結果，主要是在要求如何減少斷字 (hyphenation) 而中文字每行可容納的字數是固定的，因此沒有斷字的問題，因此中文小樣的排版比英文容易，

不必考慮如何斷字。中文報紙的組版就有點類似英文的排小樣，然而其複雜程度比英文排版有過之，因為如果把一段新聞當做一個英文字時如何剪切非方塊新聞以填入二維空間，比如何切斷英文字的一維 (one dimensional) 情形要複雜。

我們先訂出中文報紙排版的一些規則及限制 [13]，從而模擬出定義版面美觀與否的測度。

1. 版面大小的限制

一般中文報紙而言，全版縱向有 21 欄，每欄為 9 個字，橫向有 115 行左右。但因有廣告佔去版面，真正新聞版面大小為可變固定值，而新聞稿必須填滿整個版面。

2. 二篇小樣的正文不得在同一欄中緊接著

也就是說二篇小樣的正文如果在同一欄出現，中間必須有標題分開。

3. 每篇新聞的空白行數限制

在組版的過程中，某些小樣可能會產生一些無法填滿的空白行，這些空白行有時是無法避免的，如圖 3-1 所示。空白行太多不美觀，所以應該盡量減少空白行。在同一小樣中的空白行數有一限制不得超過一固定值，一般設定 3 行。小於 3 行者可以容忍，超過 3 行即視為不可。

4. 小樣標題重疊寬度限制

每篇新聞都有標題，在組版過程中，編輯均不允許二篇小樣的標題重疊部份太多。所謂重疊乃是指一篇小樣的標題的底端和另一篇小樣標題的頂端相聯，如圖 3-1 及 3-2 所示，圖 5-1

美共和黨裔族委會

未設榮譽會員職位

朱兆祥在台設分會一事
亞裔總會決定提出質詢

加國軍運裝備

中共有意購買

孫院長勗勉中華成績

亞洲杯再創輝煌戰績

【本報訊】中華全國體育協進會，日前在東京舉行的亞洲杯足球賽中，中華隊以三比零擊敗日本隊，獲得冠軍。孫院長在賽後表示，中華隊在比賽中展現了極強的團隊精神和戰術配合，這不僅是球員們辛勤訓練的結果，也是全體中華同胞支持與鼓勵的體現。他勉勵中華體育事業在未來取得更大的成就。

壽星團出「百集令」

救日高手「遠歸隊」

【本報訊】由各界人士組成的「壽星團」日前舉行了一場別開生面的「百集令」活動。活動中，多位壽星在眾人的歡呼聲中，展示了驚人的「救日高手」技能。這些技能包括在極短時間內完成複雜的動作，展現了他們非凡的體魄和敏捷的反應。活動在歡樂的氣氛中圓滿結束。

亞洲杯我將三屆冠軍

【本報訊】在東京舉行的亞洲杯足球賽中，中華隊以三比零擊敗日本隊，獲得冠軍。這是中華隊在亞洲杯賽中取得的第三個冠軍，也是隊史的最佳成績。隊員們在賽後表示，他們感到非常自豪和榮幸，並將繼續努力，為中華體育事業做出更大的貢獻。

美軍運「動作迅速」

中華隊「忍無可忍」

【本報訊】據悉，美國軍隊在運送軍用物資時，動作非常迅速且效率極高。然而，中華隊在面對此種情況時，表現出了極大的忍耐力和韌性。儘管面臨種種困難和挑戰，中華隊始終堅持不懈，展現了中華民族不屈不撓的精神。

（此處為圖 5-1 的說明文字，內容與上方新聞標題及內容相關，但文字較為模糊，主要描述圖中表格的結構特點。）

圖 5-1 含有空白行的小樣

圖 3-2 標題重疊

顯示二個標題重疊寬度為 2 行，仍然可以忍受，圖 3-2 的標題重疊部份太多，違反了組版美觀的原則，通常是不允許的，除非真的無法解決，才會出現這種版面。有一種情況不在此限制內，即二篇小標題若有一個標題為橫標題，則不計其重疊部份。

版面的好壞除了要符合以上的限制之外，必須滿足一些其它的美觀條件。

- 1 每篇新聞的分段個數不得太多，每一小段之間的距離不得太遠。
- 2 愈重要的新聞必須愈接近右上角。
- 3 標題不得平行。
- 4 標題的欄位至少要合幾行正文。

把這些限制及美觀條件用數學的形式表現出來，則尋求最佳版面的問題就是求解下面的作業問題。

“選取非方塊新聞及方塊新聞，將新聞小樣填滿版面，在填的過程中如何將非方塊新聞分段及安排小樣的位置，使得處罰評分為極小值”。選取新聞 L_1, \dots, L_n 滿足

$$(L_1 + \varepsilon_1) + (L_2 + \varepsilon_2) + \dots + (L_n + \varepsilon_n) = \text{版面大小}$$

使處罰評分

$$f = \sum_{i=1}^n [C_i^1 * W_1 + C_i^2 * W_2 + \dots + C_i^5 * W_5] \quad \text{爲極小。}$$

For $i = 1, 2, \dots, n, j = 1, 2, \dots, 5$

L_i : 新聞大小 (長度)

ϵ_i : 容許變化 (空白行或縮小標題使得新聞長度有加減的變化)

C_i^j : 分項評估值

C_i^1 : 分段個數

C_i^2 : 空白行數

C_i^3 : 標題重疊行數

C_i^4 : 標題平行

C_i^5 : 標題欄位含太少正文

W_j : 加權函數 (Weighting factor)

W_i 可能為 C_i 的函數，譬如說每篇小樣的空白行數超過 3 行時即 C_i 值大於 3 時則加權指數 W_i 就遠大於小於 3 行的加權指數。

有了這個數學模式，我們可以有一個指標去尋求美觀的版面或最佳版面。

最佳版面的尋求非常的複雜，因為它不只是一個新聞排列順序的問題，而且常涉及每一新聞的位置問題。它的複雜程度不是用窮舉法 (exhaustive search) 所能解決的。如果採用機動作業 (dynamic programming) [4] 的方式可以降低複雜度，但是資料結構，作業流程的安排仍然十分複雜，還在研究階段。

另一種比較可行的方式是不求得最佳解，但是找到一個可以接受的美觀解，也就是說只要找到一個版面形式，它的處罰值不

超過某一臨界點，則這樣的版面達到某一程度的美觀，可以符合出版的要求即可。下面一節，我們就簡單介紹這套軟體系統的結構及作法，詳細的資料可參閱[5]。

3-3 啟發式的組版系統

此一組版系統，能綜合組版規則與一些編輯組版經驗，找出一套具有啟發性(heuristic)方法，做到智慧層次，盡可能的組出最佳版面。

整個排版系統分兩個組模，一個是小樣編排組模[4]另一個就是這裏介紹的組版組模。此一組版組模是利用小樣編排組模產生的新聞小樣資料檔來組版，這個小樣資料檔包括組版時所要的小樣特性資料(1)，新聞種類方塊或非方塊(2)新聞大小，(非方塊包括標題大小及內容長度)(3)新聞的重要程度，組版過程會產生一些中繼檔案(intermediate file)，中繼檔案是組版系統在作業過程中所產生的一些暫時檔案，這些檔案主要用以配合倒回搜尋法的使用達到自動組版的效果。完成後的版面以一個媒介語言所描述的媒介檔，配合小樣組版產生的媒介檔可以產生完整的版面。其結構設計如圖3-3所示。

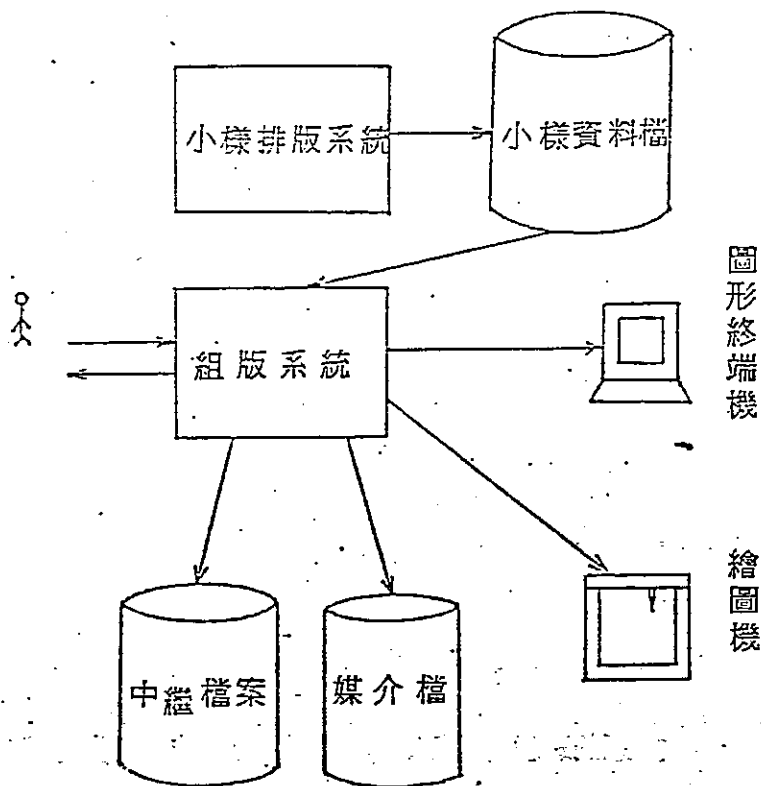


圖 3-3 自動組版系統結構圖

組版之模倣作業方式

組版系統採用的是自動與交談式混合作業，作業模式如圖 3-4

所示。

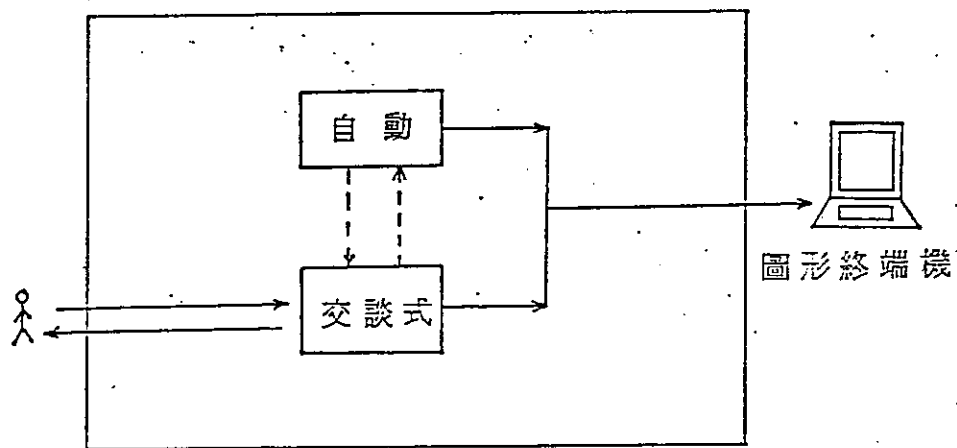


圖 3-4 系統作業模式

表示資料傳輸 —————>

表示控制權轉換 - - - - ->

自動作業與交談式作業之間的關係為：

a. 自動----->交談式

自動作業將控制權轉交給交談式作業的情況有二：

- (1) 當自動作業遇到無法處理的情況時，例如某段新聞的標題太大，正文太短，或其它原因使得自動作業無法再繼續處理其它新聞時，自動作業便會把控制權轉交給交談式作業，讓使用者做版面的修改，達到組版的目的是。
- (2) 當自動作業找到其最佳版面時（在此指版面均已填滿新聞），系統將會結束自動作業，然後把控制權轉交給交談式作業。若是使用者對此最佳版面不滿意，則使用者可以利用交談式作業做進一步的修改，達到使用者心目中的最佳版面。若是使用者覺得滿意，便可以結束交談式作業，系統便會結束作業。

b. 交談式----->自動

在某些狀況，使用者利用交談式作業做完一些修改動作之後，便可將控制權轉交給自動作業，繼續組版動作。

自動作業的流程如下圖 3-5 所示

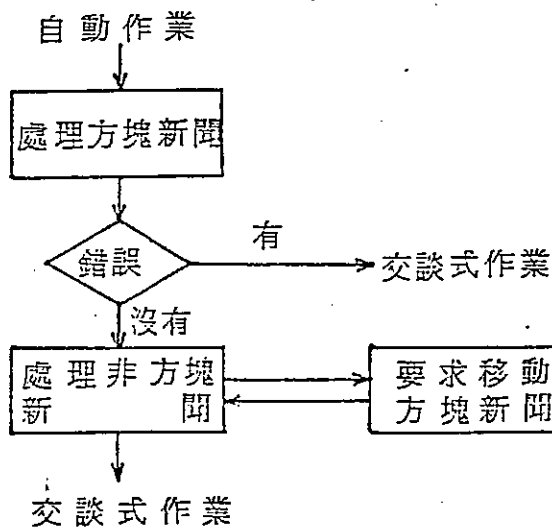


圖 3-5 自動作業流程圖

在處理方塊新聞時若發生排不下的情況，則交給交談式作業處理。處理非方塊新聞時，可以要求移動方塊新聞，達到組版的目的。最後，不管版面是否填滿，都交給交談式作業處理，以便使用者修改版面。

交談式作業的設計是爲了彌補自動作業的不足，使用者可以在交談式作業中隨意的修改版面，但仍然要遵守版面資料結構。交談式作業流程圖，如圖 3.6 所示。

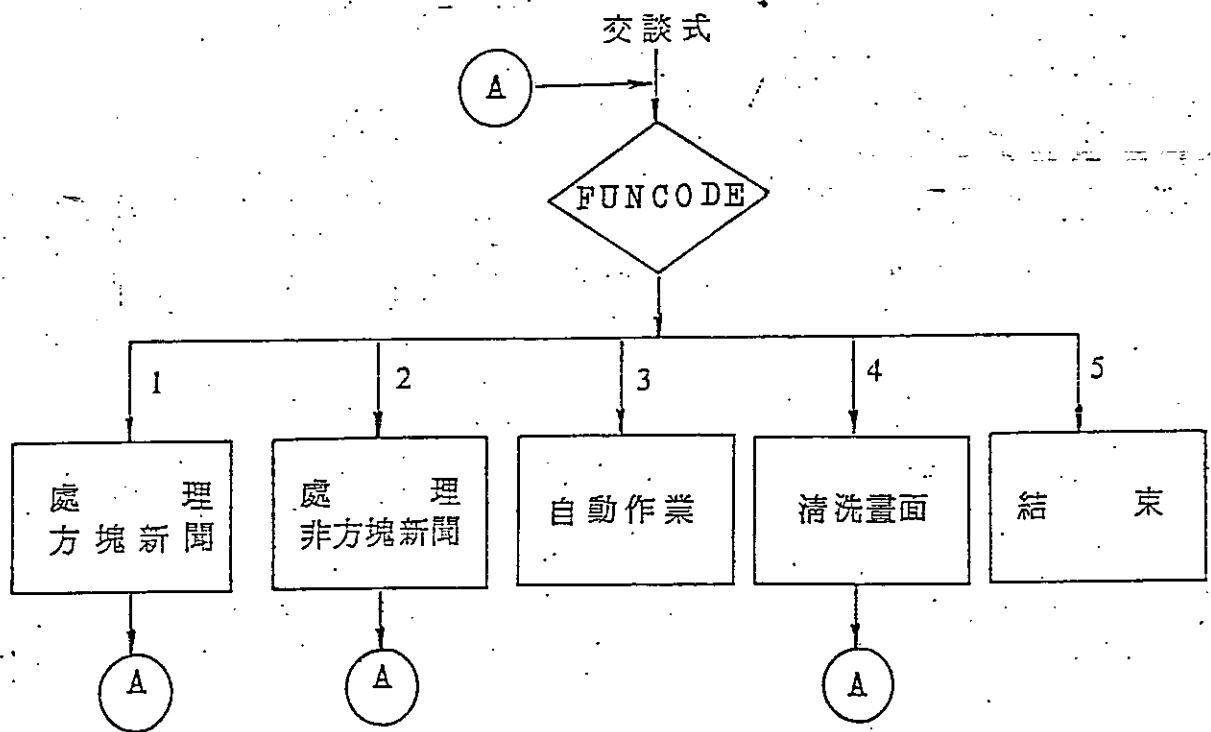


圖3-6 交談式作業流程圖

在交談式作業中，使用者有下列五種選擇：

- (1) 處理方塊新聞
- (2) 處理非方塊新聞
- (3) 回復自動作業
- (4) 清洗畫面
- (5) 結束

使用者可以依其需要，選擇一個函數碼 (FUNCTION CODE)

如果使用者選擇(1)處理方塊新聞模式，則他可以利用下述命令隨意修改所有的方塊新聞。

- (a) 移動 將方塊新聞移動至任意位置，然移動後之版面應為合理之版面，否則不得移動。所謂合理之版面為每一列上的空白行為連續，即不得有兩列空白中間為新聞分開。
- (b) 填入 將一個尚未填入的方塊新聞填入版內所指定位置。
- (c) 去掉 將版面上的任一方塊新聞去除，然去除後之版面必須為合理之版面。
- (d) 顯示方塊新聞資料。
- (e) 重新顯示最新版面。
- (f) 回到交談模組。

使用非方塊模組時，各個命令的功能分別敘述如下：

(1) 倒退命令

由於系統在處理非方塊新聞是有一定的程序，為了不破壞此程序，系統不允許使用者任意去除某一篇非方塊新聞，因為這並沒有意義。所以系統只允許去除最後排入的非方塊新聞，採用的做法為後進先出 (Last In First Out)。若使用者選擇使用此命令，則系統會詢問要退幾層：

以圖3-7 為例，假如非方塊新聞填入版面的順序為：

- ① 順序號碼為 3 的標題。
- ② 順序號碼為 4 的標題與正文。
- ③ 順序號碼為 5 的標題與正文。

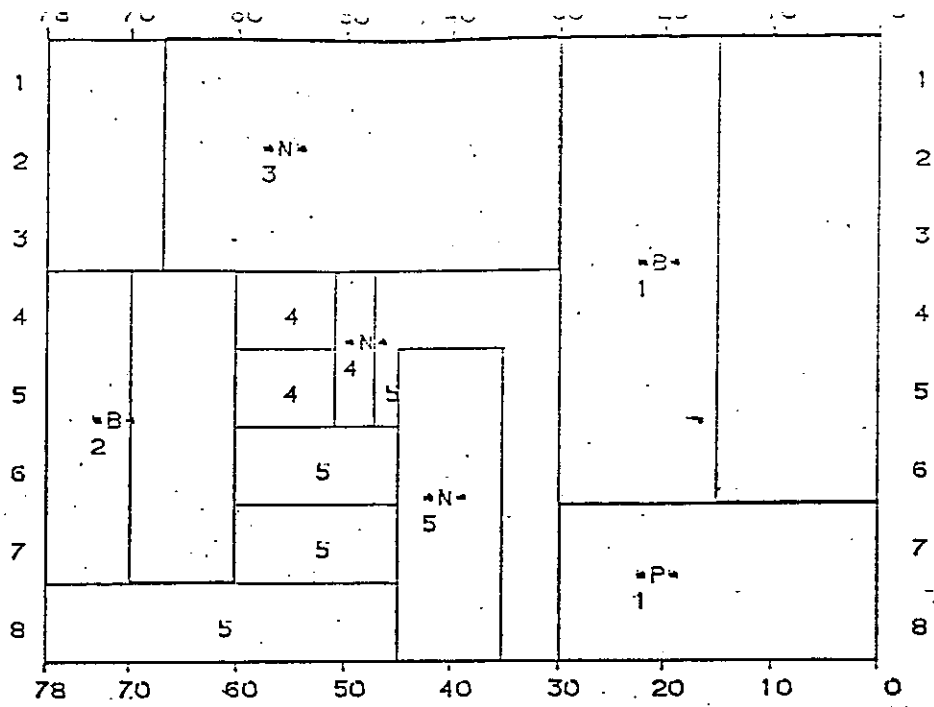


圖 3-7 說明倒退命令範圍

若使用倒退命令倒回一層，則所得到的版面如圖 3-8 所示，最後填入版面的新聞（即順序號碼為 5 的非方塊新聞），已經被提出版面。在此所要注意的是，若上一層的版面有移動方塊新聞，則回復原來版面狀態時，也要將方塊新聞移回去。

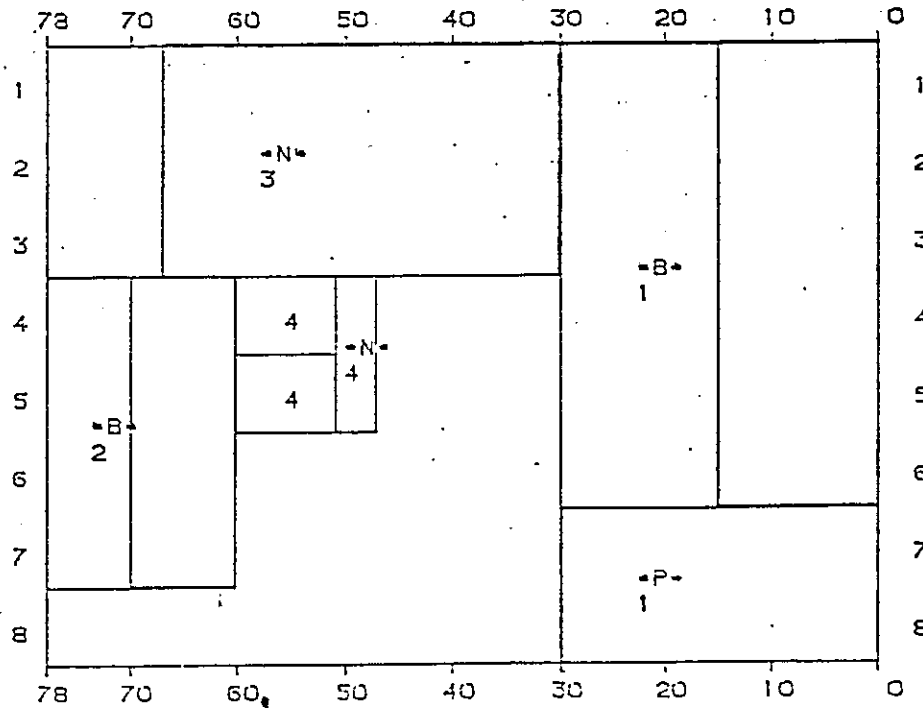


圖 3-8 利用倒退命令退回一層

(b) 分割填入命令

利用分割填入命令，使用者可以切割某一非方块新聞的正文，將其填入指定位置。在此，新聞的選擇和新聞的優先次序無關，所以給予使用者很大的選擇彈性。選擇使用分割填入命令之後，系統會詢問使用者要將所選擇的新聞放在版面的左邊或右邊。此二種選擇有很大的不同，茲以圖3-9 的版面為例，說明如下：

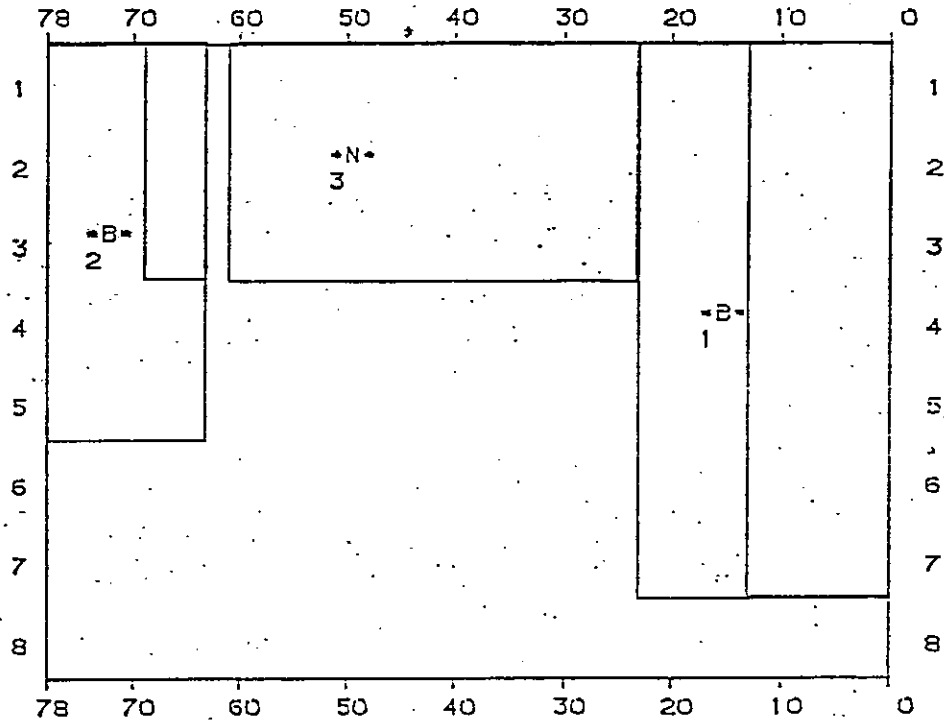


圖 3-9 說明分割填入命令的範例

a. 往右邊切割

假如使用者希望切割新聞順序號碼為 3 的正文，且將新聞順序號碼為 4 的標題放在右邊，得出如圖3-10 的版面，其中最後詢問步驟允許使用者加入一些優先次序為最低的小新聞，以便增加切割正文的彈性，因為有時會發生正文太少的現象，以致於無法切割。

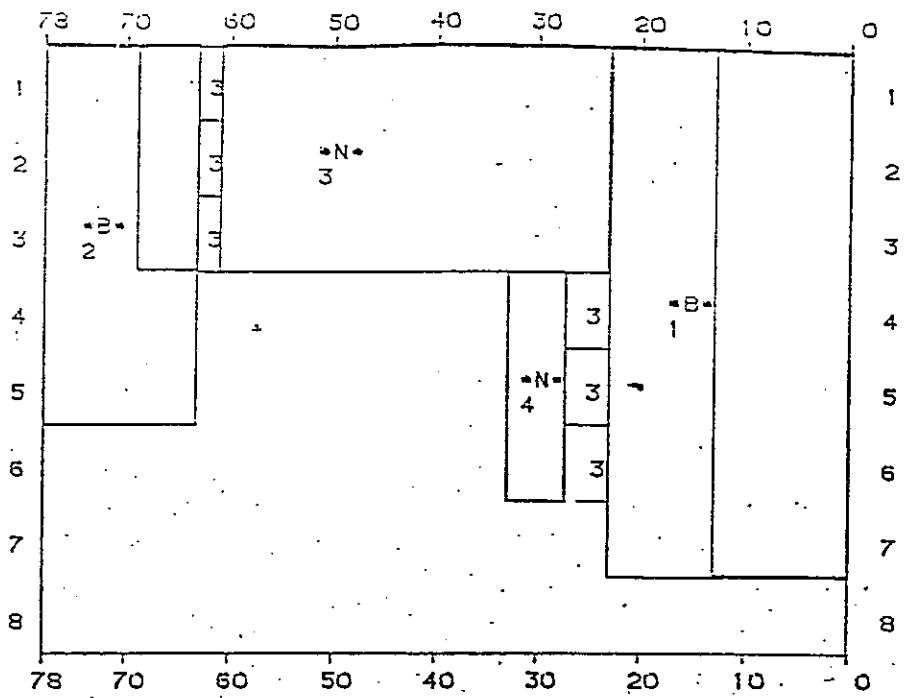


圖 3-10 往右邊切割範例

b. 往左邊切割

假如使用者希望切割新聞順序號碼為 4 的正文，並且將其標題放在左邊，得出如圖 3-11 的版面。

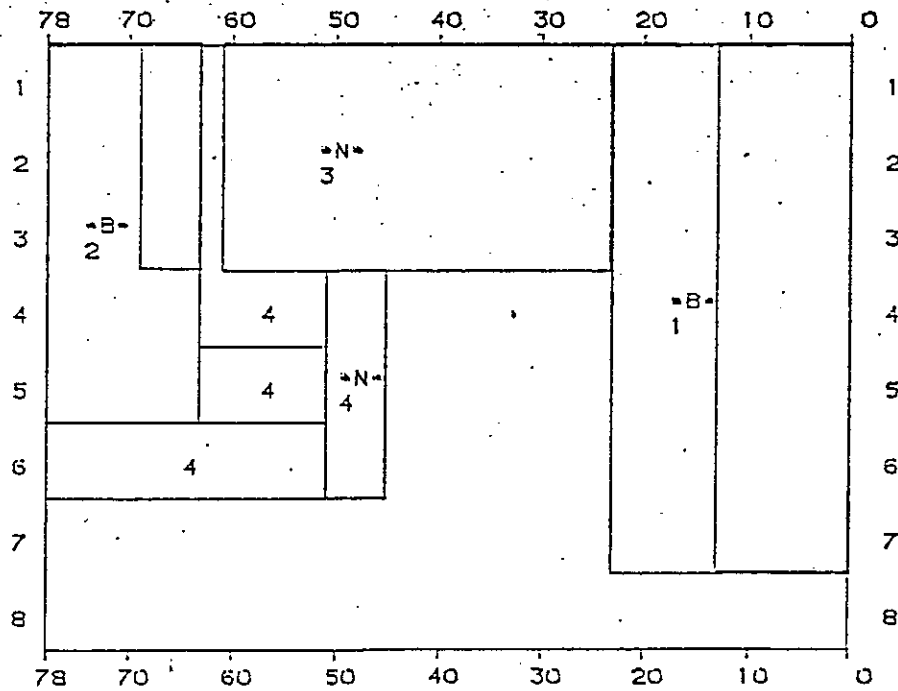


圖 3-11 往左邊切割範例

(c) 顯示所有非方塊新聞資料

(d) 改變命令

系統允許使用者利用此命令，改變系統參數：小樣中空白行數限制標題重疊寬度限制。

完成此一系統主要的方法是要訂出簡單的資料結構，使整個搜尋的過程為一樹狀結構，原始版面為樹根 (root node)，而中斷版面為樹狀結構中的節點 (node)，而完成的版面為樹葉 (leaf node)，自動作業採用倒回搜尋法的深度搜尋 (D-search)，並採用倒回搜尋法 (Backtracking) [15]，去尋得第一個可接受的美觀版面。系統的詳細設計請參閱 [5]。

3-4 結 論

此一系統雖是自動和交談式混合作業，但主要還是針對自動作業而設計，交談式作業只是輔助性質，經過實際的模擬編排，以一版的新聞來說，自動組版系統大約只要 5 分鐘，便可找出可接受版面，若使用者要修改版面，則所費時間也很短約 2,3 分鐘。因此，整個系統作業時間相當短，比起傳統的報紙作業時間，快得很多。至於最佳版面的搜尋目前仍在研究階段，採用的方法不外乎機動作業及截枝方法 (Forward Pruning) [16] 及左下往右上堆疊的方式以減少搜尋的時間及範圍。

參考文獻

- 1 R. Furuta, J. Scofield, and A. Shaw, "Document Formatting Systems Survey, Concepts, and Issues", Computing Surveys, Vol.14, No. 3, September 1982.
- 2 廖慶和, "文件處理系統—版面規畫子系統" 交通大學, 碩士論文, 1983.
- 3 陳宏渠, "文件處理系統—內容編排子系統" 交通大學, 碩士論文, 1983.
- 4 許永高, "中文報紙小樣的自動化編排模擬系統" 交通大學, 碩士論文, 1983.
- 5 廖敏芳, "中文報紙自動組版系統" 交通大學, 碩士論文, 1983.
- 6 張系國編著 "中文輸入輸出系統" 中央研究院數學所, 民國六十二年。
- 7 資訊策進會技術通報 C18 號 "中文電腦發展調查分析報告" 民國七十一年。
- 8 K.J. Chen and K.Y. Cheng, "A Structured Design Methodology for Chinese Character Fonts", Proceedings of ICTP'83, Japan, 1983.
- 9 J. C. Tsay, "Automatic-Processing System for Chinese Character Generation, Text Editing, and Typesetting" Ph.D. Dissertation, National Chiao Tung University (NCTU), 1974.
- 10 Y.J.Chen, "Computerized Chinese Typesetting System", NCTU, 1977.
- 11 I.C. Wu, "Computer-Assisted Page Make-up of a Newspaper", NCTU, Master Thesis, 1981.

12 D. Knuth, TEX and METAFONT, Digital Press, 1979.

13. 陳石安, "新聞編輯學", 軍事譯粹社。

14. R. Bellman and S. Dreyfus, Applied Dynamic Programming, Princeton University Press, 1962.

15. N. J. Nilsson, Principles of Artificial Intelligence, Springer-verlag, 1980.

16. J. R. Slagle, Artificial Intelligence: The Heuristic Programming Approach, McGraw-Hill, 1971.