PAGESETTING AND CONSIDERATIONS OF DATA FORMATS

Donald G. Gerlich, Information International, Inc.

(Slide of Full Newspaper/Magazine Page) (2)

Introduction: The title of this paper vaguely refers to the problems of manipulating a variety of electronic data formats for the "typesetting" of a complete page with both text and graphics.

(Slide of a Variety of Equipment) (3)

With the proliferation of a variety of scanners, terminals and recording devices, a major system design problem is the storage and processing time per unit of visual material as a function of the data format. The subsystem integration problem is significant.

(Slide of a Film and Plate) (4)

This paper will attempt to highlight these considerations by contrasting a system which uses symbolic data or coded data, to record on traditional CRT pagesetters, to a raster data or facsimile data system which can record on the futuristic direct plate recorders. As examples, both production and experimental systems of Information International will be discussed.

(Slide of Copy Blocks on a Terminal) (5)

The third consideration of data formats will be the interface questions between the back-end recorders and the front-end text processors, in the specific areas of fonts, set widths and line hyphenation and justification.

(Slide of Newspaper Pagination System) (6)

1.A. Symbolic Data System

The functions of a standard Information International Newspaper Pagination System are:

- The File Manager inputs and stores editorial data from the front-end text processing system which has produced hyphenated and justified stories and headlines.
- The Scanner digitizes continuous tone photographs, storing 256 levels of gray for each screen cell.

- It also scans images of camera-ready ads and stores them in run length codes. Although the scan input is limited to 8 x 10 inches the output size is independent of the input size.

- The Page Makeup Station, using a variety of stored electronic dummies, provides the ability to interactively select text or graphic components, scale and position these components, and display the completed page with true typefaces and graphics.

- The electronically made-up page is sent to a CRT pagesetter which sets a full-size broadsheet page on film or photosensitive paper.

- The completed masters are then used to make up plates in the conventional manner, or for facsimile transmission.

(Slide of Text String Codes and Halftone Dot Codes) (7)

The data formats in this system, as stored after pagination and transferred to the recording unit are primarily symbolic, that is each text character is represented by one 8-bit numeric does as is the gray scale value for halftone dots.

(Slide of Stroked Graphic Arts Characters) (8)

The pagesetter seeks digitally "stroked" images of graphic arts fonts and "paints" each character on the CRT.

(Slide of ESP) (9)

The Pagesetter also analyses gray scale bytes and generates halftone dots. In general , the size of the dot will be a function of the amount of black sensed. However a procedure called ESP for enhancement sharpness procedure, samples a neighborhood of gray values and modifies the nominal dot value.

(Slide of Full-Face Page With Sequence of Recording) (10)

Any pre-screened art, which has been stored in run length code is mixed with the symbolic data and recorded as a complete entity.

(Slide of This Table) (11)

Summary of the 3 data formats uses

	Data Type	Storage Per Unit
1.	Text and Control Codes	1 byte/character
2.	Continuous Tone Input, Halftone Output at 85 Line Screen	9,000 bytes/sq. in.
3.	Line Art/Pre-Screened	8,500 to 40,000 bytes/ sq. inch

(Slide of Magazine Network) (12)

Benefits of These Data Formats

A. Compact text and continuous tone formats afford editorial magazine page transmission in approximately 4 minutes for remote pagesetting with 9600 baud lines.

(Slide of Gray Scale Playback Curve Manipulation Effect) (13)

B. Storage of continuous tone format affords image enhancement at page setting time.

Using these formats the Typical Newspaper Storage per page is:

(Slide of This Table) (14)

Text and Control	10,500 Bytes
Halftones	660,000 Bytes
Line Art and Pre-Screen	1,080,000 Bytes
Total	1,750,000 Bytes

(Slide of This Table) (15)

vs. Facsimile Format of This Page

Text	2,115,000 Bytes
Halftones	3,000,000 Bytes
Line Art and Pre-Screen	1,080,000 Bytes
Total	6,195,000 Bytes

The ratio of symbolic data to the facsimile page data is 1:3.5. This is one indication of problems of raster recording versus symbolic recording devices.

(Slide of Text Format Expansion) (16)

2. THE PROBLEM OF RASTER RECORDING

The fundamental problem to be solved is how to organize megabits of data so that a raster recording device - using a laser with deflection in one axis - can image characters and illustrations at very high speed.

The slide shows the text problem graphically as characters move through each step of the process and the amount of data per character multiplies at each step.

Then there is the sorting problem. All the characters on a page must be sorted left to right, top to bottom. The entire page must be assembled before any recording. Illustrations, either halftone dots or line art, must be converted to bit maps and merged with the page raster. Segments of all characters or picture elements on one raster line are recorded with each sweep of the laser.

(Slide - Exploded View of a Small Area to Show Laser Recording Path) (17)

There are approximately 1,000 sweeps per inch. In order to record a 15" x 20" newspaper page in one minute, the recorder must deflect the laser 20,000 times over 15" each time while moving the material with equivalent accuracy. In terms of data, the recorder must process 30,000,000 bits per minute. Sufficient work has been done to prove the feasibility of achieving this rate.

(Slide - Picture of Crosfield System) (18)

As a matter of fact, Information International has built a device for converting symbolic graphic arts characters to variable resolution bit maps for laser recording. This device (called the DCT) is integrated in Crosfield's Pre-Press Color system, the Studio 800.

In addition to the basic logic problems there are attendant physical development concerns such as material movement accuracy, laser power and reliability, material sensitivity and reproduction life, etc. These are left to another discussion.

(Slide - NPS/Platemaker System) (19)

Now let us look at a similar system but modified to drive a laser platemaker directly. We have added a:

Text Conversion Unit Page Merge Unit
Laser Platemaker

in lieu of a CRT Pagesetter and Conventional Platemaker.

Before we examine the data format implications let's briefly review the results of the pagination process.

(Slide - Page Components) (20)

The completion of pagination results in a component file for each page which designates the exact size and location of every element on that page. This file is transferred to the Merge Unit.

(Replay of Slide 6) (21)

(Replay of Slide 14) (22)

In the symbolic system the Pagesetter requests a component - story, photo, or ad - and records <u>all</u> the data for that component and then moves to the location of the next component anywhere on the page. Thus page assembly is done in symbolic form.

(Replay of Slide 17) (23)

With a raster recording device all the data of each component of a page must be assembled in left to right, top to bottom order, which is the output recording sequence.

Only then can segments of each component be recorded across the page.

Furthermore, the page assembly must be performed on data in a format suitable to drive the platemaker. This is a bit stream, generally compacted in the form of run-lengths.

That is, the page data, at Page Merge Time is in the facsimile format, or over 6,000,000 bytes per page form. (Previously discussed).

(Slide - Replay Slide 19) (24)

Where as the CRT Pagesetter needs to fetch approximately 1.7 megabytes and could record components "randomly", the page merge unit must fetch 6 megabytes, and most importantly, sort all the components in sequential recording order, and then drive the platemaker.

For the average newspaper, this must be done under 2 minutes per page, at 800 lpi. Our design goals are 1 minute per page at 1,000 lpi.

This is a non-trivial data processing problem.

Reviewing the 3 inputs to the merge unit, they are:

1. The TCU. This unit will convert each character and halftone value which is in the input data format to a "bit string". The bit string of a character is "shaped" by data from the VideoFont library stored on the TRC. The VideoFont library is the same as that on the Pagesetter.

The character "bit strings" for each line in a text block are used to build horizontal run length codes.

The converted text blocks are transferred to the merge unit.

2. The Merge unit also receives the illustration data in run-length code directly from the scanner in the same code format as characters from the Text Raster Converter.

3. The component file or map of the page is transferred from the File Manager to the merge unit after pagination. Recording this component file the merge unit retrieves illustration and/or text data in left to right, top to bottom order, which is the output recording sequence.

   Using an output buffer which is large enough to accommodate at least one full line across the full width of the page (15,000 bits), the merge retrieves a run length, converts the bit string and "OR's" it into the buffer with the adjacent string. The contents of the output buffer are then converted back to compact run lengths which are compatible with the Platemaker.

   Double buffering is used for output speed and the process is continuous for a complete page.

(Slide - Generic System Block Diagram) (25)

3. The last area of concern about data formats but definitely not unimportant, is the interface between a pagination system with either a symbolic or raster recording system and the front-end composition system.

   Let's look at the simple block diagram of a generic system consisting of:

   - A Text Processor from one vendor and;
   - A Pagination/Recording system from another vendor.

The general functions are:

- The text processor or "front-end" will enter text, edit content, maintain a data base and perform line composition.
- The Pagination/Recording system will enter and store illustrations, perform page makeup and record a complete page on photo or plate material.

(Slide of Front Spacing Values) (26)

The first interface concern is that the front end uses the same set widths as the fonts on the recording subsystem.

The slides shows 4 manufacturers of popular typesetters, all of which uses different unit systems for character spacing.

The relative units (or character width values relative to the point size being set) are used in the h & j calculations. The relative unit system is established at the time the vendor manufactures the fonts and designs his typesetter.

The absolute spacing increment is the minimum space that the light source can be moved. It is a function of the equipment design.

Once defined, neither specification is easily changed.

Over the years it has been no problem for the front-end to load the proper setwidths for the intended recorder.

However if a variety of recorders are to be used, the front-end must:

A) load a setwidth table for each recorder;

B) and re- h & j the copy for each different recorder.

One line composition setting for multiple type recorders will not work.

(Slide - 2 Galleys of the Same Type) (27)

Next, let's take the case of one recorder simulating another recorder.

If we mean by simulation, 2 galleys of type, each with;

the same line justification
the same interword spaces
the same letter spaces
the same character shapes,

coming from two different machines and overlaying each other as to be indistinguishable, then we have a severe problem.

Looking at our table of spacing values you can see to achieve this the machine doing the simulation would have to be modified to have the same relative and absolute spacing values as the simulated machine, plus it would have to use the identical master fonts.

This is so impractical from an engineering/programming cost view, not to mention proprietary data rights, that it is impossible.

The next level of simulation which, "we will call APPROXIMATION" can provide equivalent line justification, but NOT equal letter or word spacing nor character shape.

(Slide - Replay Slide 23) (28)

To achieve this the back-end recorder continues to use its own fonts and again loads its setwidths into the front end system, but the input command code is that of the machine being simulated. Therefore there must be a new program in the recorder for each machine being "approximated".

Since this program cannot space any finer than the resolution of its own hardware there may be round off errors during "approximation", which cause spacing "uglies".

"Approximation" usually runs slower than its native mode because the software is doing extra calculations.

(Slide - Directory Magazine Pages) (29/30)

Although "Approximation" is sufficient for some jobs, all run on the same machine, it is not acceptable to a native page in a high quality publication.

(Slide - Replay of Slide 24) (31)

The final data relationship to be looked at is that between the front-end and the pagination system.

Since the process of pagination can include a re-h & j of copy to make it fit, it is desirable to have the h & j algorithms of the two systems as close as possible. One way to achieve this is to make sure the pagination system's exception dictionary is as inclusive as that of the front-end.

A companion problem to this, is that any editing by the pagination system - to content or h & j - as opposed to layout manipulation, will cause output which is different from the content of the front-end data base. Thus a subsequent revision operation would be operating on a version of data that did not reflect that prior edit.

This can be avoided either by:

A) Having the pagination system transmit all edited files back to the front end data base;

B) Or by performing all content edit operations on the front-end system terminal, which can be closely located to the pagination terminal.

4. Conclusion

We have tried to show some of the data dependent problems of system design.

- The data compaction and processing advantages of "sybmolic" data formats.

- The data expansion and system constraints for use of raster recording devices.

- The considerations of font formats and line justification procedures when interfacing "front and back" end systems.

We have attempted to highlight these issues because too many have minimized their concern; "It's only a matter of software", and caused longer than anticipated development and slower than expected performance.

Further, this has caused a credibility gap in our industry, which perhaps we can help close.