

A Framework for Digital Data Workflow in a Graphic Arts System

Robert R. Buckley*

Keywords: prepress, systems, digital, exchange, flow

Abstract: A basic desktop publishing system connects the application or creative operation to an output device via a driver and RIP. As the desktop has become a more credible source of input for graphic arts and production systems, the demands on productivity, quality, expertise and integration with existing processes has grown. One result has been a more elaborate framework for digital processing operations from creation to output, as described in this paper. Along with the proliferation of processing stages has come a growth in commercial opportunities, standardization activities and workflow scenarios. The use and application of this framework are illustrated by examples drawn from existing commercial systems.

Introduction

In a graphic arts system, a creative source is linked to a production destination. The basic elements of such a system can serve as the starting point for describing the evolution of digital data processing and flow in graphic arts systems. These basic elements are shown in Figure 1, where an application is connected to an output device via a print driver and RIP (Raster Image Processor).

The application is feature-rich editing software with a good user interface for capturing user intent and generating the desired document appearance, which is expressed in terms of an internal format. The driver converts the internal format to a PDL or page description language, such as PostScript or PDF (Portable Document Format). Most high end graphic arts applications have integrated drivers. The RIP converts the device-independent PDL into a device-dependent raster image for the target output device. The output device can be an imagesetter, direct to plate, direct to press, or a laser printer. While the physical medium can influence the overall workflow, this paper will focus on the digital data flow and operations in the system.

*Digital Imaging Technology Center, Xerox Corp., Webster, NY

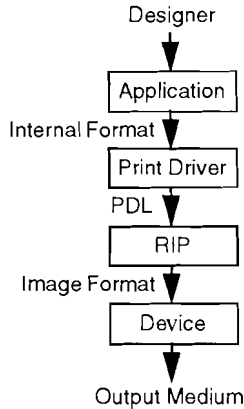


Figure 1: Basic Framework

Taken together, these steps form the core of a commercial system. Since they precede the output, they are “prepress” operations. Adding the necessary infrastructure, including platforms, networks, storage and databases would turn the framework of Figure 1 into a commercially viable system.

In this basic system, the application is the value added operation. The remaining steps are essentially format converters. They translate and interpret the content choices made by the document designer at the application into formats appropriate to the different stages in the flow. In the progression from application to output device, the ability to modify document content decreases, while data volume and dependence on output device characteristics increase. For example, the application’s internal format describes a document compactly in terms of objects, such as lines, figures, characters, images and so on. By comparison, the image format describes the document in terms of pixels that are tuned to the output device’s resolution, inks and operating behavior, possibly including compensation for characteristics such as misregistration and tone reproduction.

Multiple Applications and Outputs

While Figure 1 shows a vertical, one dimensional flow, the system designer has a choice of applications and output devices, depending on the nature of the job, the destination and throughput requirements. The system designer may configure the system to offer multiple applications, with integrated or shared drivers, and multiple outputs, each usually with a dedicated RIP (Figure 2).

The enabler for such a horizontal, two dimensional system is the PDI., usually PostScript, which is the common intermediate format that a variety of different applications can use to connect and communicate with a variety of

different devices. Using PDL theoretically decouples applications and output devices, so that an individual application or device can be added, updated or deleted without affecting the rest of the system.

One of the output devices in Figure 2 would be used for production, while another would be used for proofing. In this case, the designer (or operator in the prepress environment) runs through the steps to a proofer, whose output provides feedback, indicating where changes and corrections are needed, and something for the client to approve, before committing to a production run. Presumably the proofer's output is an inexpensive, accessible simulator of the production output.

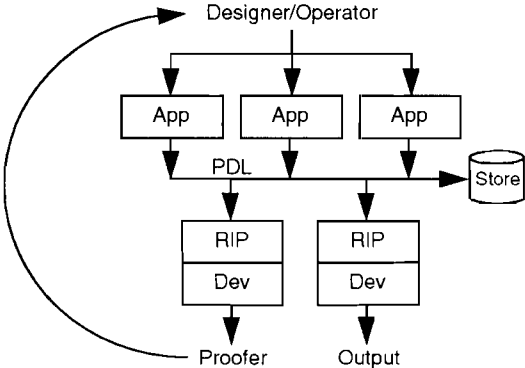


Figure 2: Multiply connected system

The proofer is part of a feedback loop shown in Figure 2 that encompasses the entire process from application through RIP to output device. It is desirable to make the feedback loop tighter and faster, either with soft proofing, by having the proofer co-located with the designer or client, or by making the proof-correction cycle encompass fewer steps in the process. The latter implies that the ability to make corrections to the digital data exists other places than the application. This is especially important in the production environment, which is typically separated from the designer and the application, so that it is awkward and time consuming to have to go back to the application every time a change is needed.

Related to the idea of feedback and correction is the notion of binding. Binding is making and committing to output-specific choices in the digital data content. Examples of binding choices are which inks to use, which is determined by the target output device, and what text to insert, especially when the text is variable data.

Early binding means decisions, selections and corrections are made in the application or driver, as part of the design stage, where the most skill is usually

available. An example of early binding is specifying traps in an application, such as QuarkXPress. In the basic digital data flow shown in Figure 1, there is often no other place to make some binding decisions. Early binding can make RIPping a simpler process, but it makes the PDL file more output specific and less portable. This means that redirecting a job to a different output device happens back at the application.

Late binding delays output-specific decisions to the last possible moment. This improves PDL device independence and portability. It also means that modifications do not require going back to the application and in effect starting over, which would reduce productivity. But late binding places greater demands on the capabilities and skills needed later in the digital data flow. Besides where one does the binding, how one does it is also an issue, and specifically whether it is interactive or automated.

Prepress Systems

The basic system in Figure 1, typical of desktop applications, has evolved into the more elaborate framework shown in Figure 3, which applies to prepress systems in graphic arts applications. The transition from one to the other has added operations before, after and within the RIP. This means added opportunities in the system for changing content and adding value to the final printed product. Several forces have driven this transition:

- Improved productivity
 - greater throughput
 - faster turnaround
 - more predictability
 - reduced rework
 - more automation
- Increased connectivity
- Improved quality
- Reduced cost
- Enhanced capabilities

Improved productivity has several components, from getting a job through the system and into the client's or customer's hands more quickly, to being able to having the system operate in a more predictable or reliable way in the face of the vagaries of input and output. Other ways of improving productivity are reducing the rework and backtracking when something does not come out right, or does not come out at all, and looking for places where processes can be automated, thereby reducing the need for interactions and decisions, which take time, energy and skill, and increase cost.

There is the continual demand to reduce cost, improve quality and meet customer requirements and expectations. Another force is improving connectivity on both the input and output sides. Enhanced capabilities and new features create new product opportunities and services, attract business and provide an outlet for innovation.

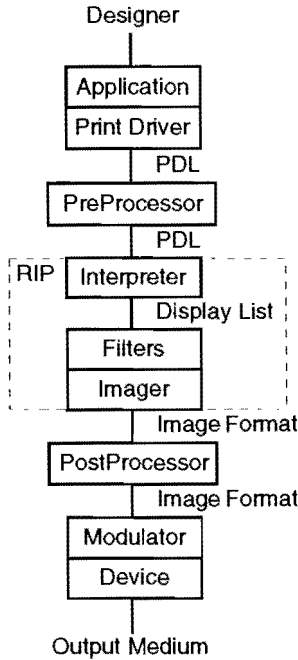


Figure 3: Enhanced Framework for Prepress Operations

Pre-RIP Processing

When files are transferred from the design to prepress environment, there are often problems, such as missing fonts, usually due to a lack of coordination between the two environments. One wants to catch these problems early and not discover them after they have taken up time, capacity and materials on the output device. It is also desirable to fix them locally, without having go back to the designer. Besides, they may not be design problems in the first place.

Some of these problems can be handled with a preprocessing step, in this case preflight, which analyzes and assesses the PDL file before it is committed to output. Preflight either fixes errors in the file or alerts the operator in the prepress shop, who can go back and fix things at the application or else make sure the appropriate downstream resources are available.

Another reason for preprocessing is performing needed operations, such as trapping and imposition, which the RIP may not be capable of doing. Luminous TrapWise and PressWise are examples. They take in a PDL file, or multiple files in the case of imposition, and output a single file that describes a trapped page or signature and that can then be sent to the RIP.

An OPI (Open Prepress Interface) server is another example of a preprocessor. It takes in a PDL with the appropriate structuring conventions and a low resolution scanned image used for placement and layout, and replaces the latter with the high resolution version. This minimizes the size of the PDL file at the source and saves time in its transmission, always useful but especially so when the transfer occurs between two different sites over a low bandwidth link. It also takes advantage of existing prepress skills and experience in preparing images for printing, thus avoiding the need to replicate those skills at the design stage or in the application.

Some preprocessing operations are candidates for integration with the RIP. Adobe has recently announced trapping and OPI as PostScript Level 3 RIP functions.

In-RIP Processing

The next place to add capabilities and features is within the RIP. The RIP's essential function is to convert the PDL representation of the job or document into a raster image description. A RIP is typically split and implemented in two pieces: an interpreter, which converts the PDL description into a display list, and an imager, which converts the display list into a raster image. The display list is an intermediate representation, more suited for conversion, usually with hardware assist, to a raster image than the PDL operators.

The display list offers another interface for inserting value-add operations or "filters" into the digital data flow. Several vendors have described a display-list-based interface within the RIP: Linotype Hell's Delta List, HyWay Ferranti's GList, Harlequin's Display List Technology, and DaiNippon Screen's Page Object Module. A published display list interface offers third party vendors the opportunity to insert solutions. Among the display list filters that have been described or suggested are ones for trapping, vignette detection, OPI image replacement and page editing.

Post-RIP Processing

Another way to add capabilities to the system is with a postprocessor after the RIP. The postprocessor operates on a raster image in a specified image format.

Examples of postprocessing operations are trapping (again), imposition, spatial resolution conversions and color calibration. While the RIP can do

screening (or halftoning) and output a binary, halftoned image, screening is often a postprocessing operation in high-end applications and is implemented by the modulator which drives the output device. Placing the screening operation at the end means that the image format is continuous tone (contone).

The image format is key to enabling postprocessor operations. The format could be a simple page-oriented raster one, such as TIFF, or a more sophisticated format such as Scitex CT/LW or TIFF/IT. At this conference two year ago, Venable et al. (1995) described Structured Images as a resolution and device-independent compound raster image format. Earlier this year, Agfa announced Intellipac, a compressible image format, as part of its Intellistream output manager for processing, transferring, and storing files.

Improving Throughput

Figures 4 and 5 show two system configurations enabled by pre- and post-RIP processing in a graphic arts system. Figure 4 shows multiple RIPs for increased throughput on a single output device. This approach avoids the bottlenecks created by a single RIP or by a single page in a job. The preprocessor distributes the pages in a job across parallel RIPs. A postprocessor then collects and re-assembles the images output by the RIP and sends the completed job to the output device.

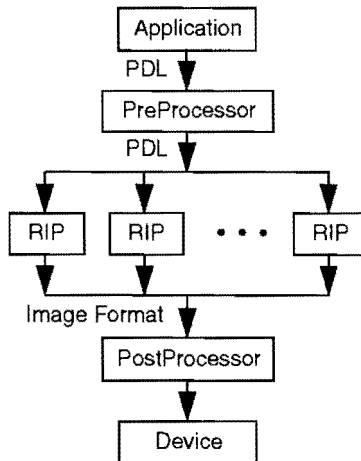


Figure 4: Multiple RIPs, single output

The multiple RIPs and single output in Figure 4 is a general description of Adobe's PostScript Extreme architecture. The page-parallel RIPping is enabled by Adobe's Portable Document Format (PDF). PDF is a derivative of PostScript. PDF has fewer operators than PostScript and is more self-contained, relying less

on external resources than PostScript does. PDF also has page independence and other document-oriented features not found in PostScript. The simpler, more rigid structure of PDF means it is less complex and less flexible than PostScript, which leads to more predictable and greater throughput.

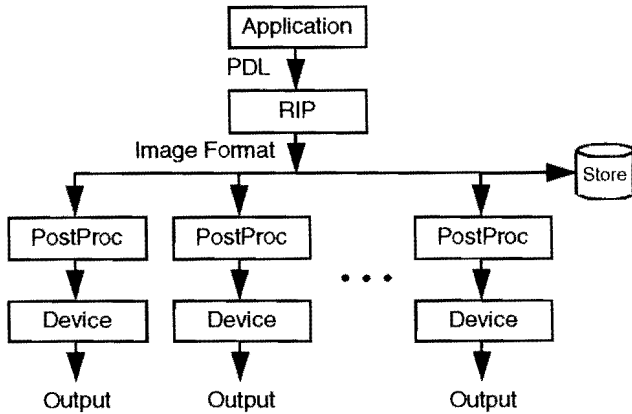


Figure 5: Single RIP, multiple outputs

Figure 5 shows another approach to increased and more predictable throughput, this time in the context of a single RIP, driving multiple output devices. The RIP may be off-line and not directly connected or dedicated to any one output device; hence, Figure 5 also shows a storage element. As a result, the output image can become an “asset” to be managed in the workflow. The output devices can be identical, in which case there is load balancing; distributed, supporting the distribute-and-print model; or truly different, in which case they all use the same RIP. In other words, the file is RIPPed once for output on multiple devices.

A single RIP and the output image it produces provide a common reference point for multiple output devices. This eliminates the variability that can occur between the outputs of different RIPs, as not all RIPs interpret a PostScript file the same way. It also enables rework and reprinting without re-RIPping, which can be important in workflows where the application or PDL file is not available or easily accessible, where the goal is simply to save time, or where the proofer uses the same image file as the production output device.

Postprocessing plays an important role in the approach shown in Figure 5. It prepares or converts the image file for an output device that may have a different resolution or color calibration than the device that the RIP targeted and that the image was originally prepared for. In effect, postprocessing enables late binding.

Image Formats

In systems, such as the one shown in Figure 5, that store, process, transmit and distribute raster images, the raster image format can have a significant effect on the postprocessing capabilities, and consequently on the workflow. In this respect, it is worthwhile briefly describing TIFF/IT and also TIFF-FX, a new TIFF-based image format proposed for Internet Fax. What they have in common is both are image exchange formats for representing pages that contain a mixture of text, graphics and pictures.

TIFF/IT (ISO, 1997) has its origins in the output of color electronic prepress systems or CEPS, which are raster oriented. It was standardized by ISO TC 130/WG2 (Graphic Technology/Prepress Data Exchange). TIFF/IT is intended for the digital distribution of color ads, which may contain a combination of pictures, graphics and text, and which may or may not be trapped. Commercial RIPs are available today that output TIFF/IT images.

A TIFF/IT file can contain multiple components: contone (CT) images, line art (LW) images, and high resolution contone images (HC). Each is encoded differently: the CT component is uncompressed at a lower resolution than the other two components; the LW component has the highest resolution and is mapped to 256 values and run-length encoded; the HC component is effectively CT with the resolution of LW. The primary color space for TIFF/IT is CMYK, although CIELu'v' is among the available options.

To combine overlapping multiple components into the final image, TIFF/IT uses a simple imaging model: LW takes precedence over CT, except when the LW value indicates transparency, in which case the underlying CT component or the paper is visible. Details are given in the TIFF/IT specification.

The International Telecommunication Union (ITU), which sets telecommunications (including fax) standards, recently dealt with a similar situation. Separate ITU standards already exist for compressing and transmitting black-and-white text and line art, color graphics, and color pictures. However, for a page that combines all three, a fax system would be forced to pick a compression method, spatial resolution and color representation that would be a good choice for one of the three, but a poor choice for the other two.

The ITU solution is the Mixed Raster Content (MRC) model, scheduled for final approval in October 1997 (ITU, 1997). Rather than describe new compression methods, MRC describes a 3-layer model for representing the scanned image data from a page that contains text, graphics and images, in either black-and-white or color. This way, each kind of content can be represented in way most suitable to it, independently of the other two and using one of the existing ITU coding methods.

In March 1997, Adobe Systems and Xerox (McIntyre and Zilles, 1997) submitted a joint proposal for an Internet Fax file format called TIFF-FX (FX stands for “fax extended”) to the IETF Internet Fax Working Group. This proposal, based on the current TIFF specification, adds the fields and values needed to represent the data produced by the suite of ITU Recommendations for black-and-white and color facsimile. In particular, the proposal enables the ITU Mixed Raster Content model in TIFF.

The example in Figure 6 illustrates the MRC model applied to a page with colored text and a picture. The 3 layers of the MRC model are Foreground and Background, which are both multi-level, and the Mask, which is binary. Pixels in the Mask layer act as a switch, selecting pixels from the other two layers for the final image. When the Mask layer pixel value is 1, the corresponding pixel from the Foreground layer is selected; when it is 0, the corresponding pixel from the Background layer is selected. Details are given in the Information and Background section of the MRC Draft Recommendation (ITU, 1997).

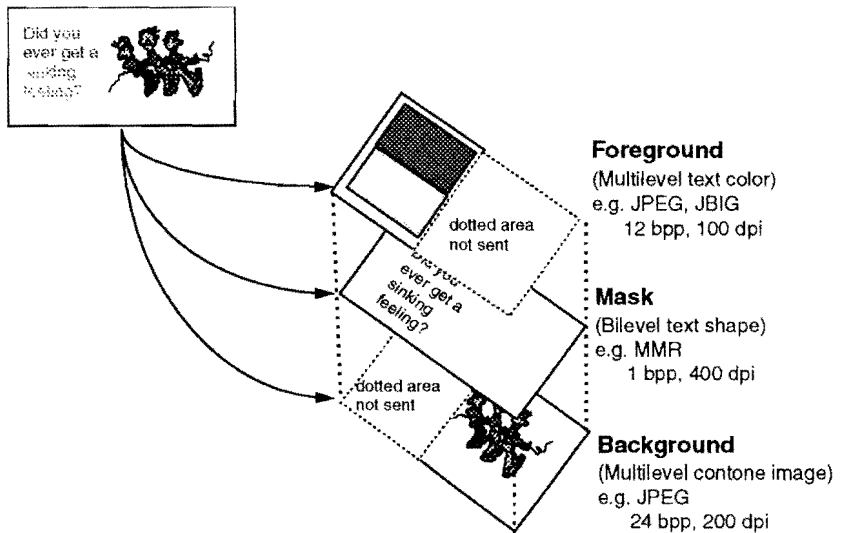


Figure 6: ITU Mixed Raster Content (MRC) Model

Each layer is coded independently of the other two, with a different choice of compression, color representation and spatial resolution. The choices shown as examples in Figure 6 are typical of a fax application. The Mask layer is a binary image, typically at high resolution and losslessly compressed. The Foreground and Background layers are usually at lower resolutions, which are integer factors of the Mask resolution (see Figure 6). The MRC model allows lossless compression

for the Mask layer and lossy or lossless compression for the other two layers. The primary color model for the MRC model in fax applications is CIELAB.

For the colored text shown in Figure 6, the filled text shapes are in the Mask layer and the text color is in the Foreground layer; the corresponding part of the Background layer is blank. Wherever, the Mask layer pixels are black (value 1), the Foreground color pixels are used for the final image. An alternative for this example is to place the colored text in the Foreground layer, and use a black rectangle in the Mask Layer to select it. Either way, the Mask layer would be coded using one of the ITU standard lossless binary compression methods, such as MMR (Modified Modified Read), and the Foreground layer would be coded using the ITU standard for lossy (JPEG) or lossless (JBIG) color compression.

In this example, the picture is placed in the Background layer; the corresponding Mask layer pixels are white (value 0), so that the Background is selected for the final image. The Background layer could be coded using the ITU standard for lossy or lossless color compression.

The example in Figure 6 illustrates the use of all three layers in the MRC model. Other pages may use fewer layers. For example, a page containing a picture would use only the Background layer. A page of black-and-white text would use only the Mask layer, with the Foreground layer defaulted to black and the Background, to white. A page with text and a color bar chart could be represented by a combination of the Mask and Foreground layers.

TIFF/IT and the TIFF-FX proposal are different solutions to representing mixed raster content documents. The differences are due to their different origins, requirements and applications. However, there is an overlap from which both might benefit when future extensions are considered. With this in mind, an informational package on the ITU MRC model and the TIFF-FX proposal was requested and distributed at the TC 130/WG2 meeting in Chicago in May 1997.

Conclusions

This paper has described the progression from the basic framework for digital data operations shown in Figure 1 to the more elaborate framework shown in Figure 3. This progression has led to a proliferation of processing opportunities, as there are more places to insert functionality and change content, both early but especially later in the flow.

The growth in processing opportunities in turn has led to growth in commercial opportunities (more elements in the system and more product choices for each element), standardization activities (more formats to keep track of, plug into and exchange between elements supplied by different vendors) and workflow scenarios (more ways to combine, arrange and connect elements; more than one place in the flow to perform a function).

All this has significant consequences for both the system designer and system manager. Not all commercial prepress systems today would have all the elements or need all the functions shown in Figure 3. The system designer would choose elements and, for those elements, the product offerings appropriate to the work process, job requirements and market served.

Given the multiple places where the file and its content can be touched and modified, another consideration is the choice of file to archive for retrieval and reprinting later: the application file, the PDL or the raster image output by the RIP. Again, the choice depends on the work process and job requirements, and would be made on a case-by-case basis.

The digital data operations in Figure 3 form the framework for developing and describing different workflow scenarios. Coordinating the possibilities for the digital data operations calls for workflow management that can accommodate different elements, file formats, databases, platforms, networks, media, job requirements and user goals.

Acknowledgments

This paper has benefited greatly from discussions with many of my Xerox colleagues, who helpful comments and inputs I would like to acknowledge.

Literature Cited

- ISO, International Standards Organization
1997 ISO DIS 12639, Tag Image File Format for Image Technology (TIFF/IT)
- ITU, International Telecommunication Union
1997 Draft Recommendation T.44, Mixed Raster Content (MRC)
- McIntyre, L and Zilles, S.
1997 "File Format for Internet Fax," IETF Internet Draft available at <http://www.ietf.org/html.charters/fax-charter.html>
- Venable, D, Buckley, R. and Yamada, T.
1995 "Managing and Representing Workflow in Prepress Applications," TAGA Proceedings, pp. 373-385

The product names mentioned in this paper are or may be the trademarks or registered trademarks of their respective owners.