 Copyright 1998, IS&T

Digital Printing Front End Systems: A Case Study of a Large Scale Variable Field, Hybrid System

William J. Ray, Ph.D.
Group InfoTech, Inc.

Introduction

This paper describes a software application program suite called Page Level Automation (PLA) that controls a hybrid conventional/electrostatic/inkjet variable field printing operation which generates very large volumes of printed material. The application suite produces between 50 and 60 million 48 page color offset booklets per week with hundreds of black plate page variations, about 2 million fully variable documents via electrostatic printing for direct mail per month and soon will produce large numbers of offset to inline inkjet variable pieces as well.

PLA allows creative users to employ those PostScript based tools that are familiar but provides a central store and file homogenization for a more fail safe production use. PLA is the first in a series of planned steps that will allow the end user to provide a true computer integrated manufacturing system (CIM) approach to both large scale print production and print production which employs highly variable field output.

The PLA Hypothesis

Risk and error are accentuated in PostScript production—especially so in the case of large format production. Given the fact that we desire to reduce such risk it is wise to consider the past and how data were controlled before the telescoping effect of modern PostScript production.

We can use this wisdom of the past to good effect in the new large scale workflow of the present. First, we start with a database. The author has argued for centralization of data in the past (PIA Technology Trends Advisory, March, 1996). *Nothing is possible without rigorous control of data.* However, most existing data scheme products are effectively stand alone image database products and, as such, have distinct limitations in practice.

Any new system should exhibit several features that should be minimal features in any subsequent product or CIM level systems integration. First, such systems should be something beyond simply an image database. The entire prepress data set and those production metadata elements associated with such prepress data need to be an intrinsic part of a CIM product. Further, tracking systems need to be built into the CIM process as intrinsic elements as well. Such tracking should include not only process state tracking but time on process tracking as well.

Ideally, such a CIM needs to be able to deal with multiple data sources *at the sub page (or cell) level* directly and seemlessly. So, PDF data, TIFF/IT data and EPS data should be intermixable within some form of geometry. Such data should then be capable of modification until the very latest possible time prior to print date — thus late binding.

Finally, a new CIM needs to provide sufficient validation of data — ideally via multiple, small granularity feedback loops — so as to prevent production until a very high likelihood of correctness is achieved.

Potential Workflow Models

To implement any CIM we need some sort of logical model on which to base a design. Two logical models present themselves, these are the database modeled workflow and the process modeled workflow.

The process modeled workflow is best illustrated by "home brew" scripting techniques, Imation's OPEN and Scitex's Brisque. All of these systems are, essentially, scripting metaphors that tie scriptable application processes together to yield an "automated" set of process steps. These techniques are powerful in the sense that process elements that require no user interaction can be automated with a simple object oriented tool.

At first blush (at least) these toolsets appear to be ideal for many prepress workflows. However, problems that arise from these systems are legion and are not always obvious. The process model workflow is intrinsically data weak, i.e., it works on the level of the application process file granularity. This means that the ability of such workflows to scale up into large, complex tracking workflows is severely limited due to the difficulty of managing large amounts of data — essentially manually.

Further, process model workflows are not intrinsically parallel workflows as they are event driven e.g. one file finishes processing and is passed along to the next process. It is difficult enough to represent parallel workflows within the GUI object model — much less actually control such workflows as some form of active process needs to coordinate and synchronize process workflows.

These workflows are also, by fundamental nature, tightly coupled in the sense that one process must lead to the next in any given schema. Two interesting consequences

arise from such tight coupling. One is that these are direct, dynamic links and, if a link is broken, the entire workflow collapses as the process chain is broken.

Secondly, process workflows are susceptible to severe bottlenecks (queuing theory easily predicts this) that are not necessarily the product of broken queue chains. So, seemingly non-deterministic events (such as how long it takes to trap a file) will hold up an entire job as the systems cannot automatically decide to either branch around a problem or parallelize a task.

So, it would appear that process model type workflows are, at best, tools to be used for specific, well understood task chains within a larger, more robust workflow.

The data modeled workflow, as illustrated by Page Level Automation (described below), Architypes MediaBank to an extent and, to a much less extent, the various image database products currently available. Data modeled workflows are characterized by their absence of explicit flow control mechanisms (such as the OPEN icon based workflow).

Work within the data modeled workflow moves through the process chain by the change of data status. So, as illustrated in Figure 7, the process light (Rdy column) changes color as data elements appear or are altered for use.

One event triggers the next not by direct process completion flags (and therefore the large granular limit which that imposes along with the tightly coupled restriction) but via the database state change — allowing finely granular, decoupled control.

This also allows a natural branching and parallelism of the workflow to occur and to be closely controlled. Workflows need not be strictly and tightly "scripted" in front but can be naturally adjusted to unplanned capacity constraints or increases.

Indeed, it can be argued that process based scripting systems need to be employed as point tools within a data modeled workflow. Thus, a data change can trigger a process event chain which will accomplish some "black box" function (as an example, trapping) which operates via a binary success or failure and on a finely granular basis — e.g. on a cell or page basis rather than a document basis.

It is the authors opinion that, in practice, process modeled workflows are not capable of being used in any other role than that of an adjunct tool within the workflow as they are logically flawed and not capable of scaling to complex tasks.

Page Level Automation

Figure 1 illustrates the general workflow of the new Page Level Automation (PLA) product. This is a real product that helps produce (among other things) four color free standing insert (FSI) booklets that are delivered with each Sunday newspaper in North America. Such FSI work, while not individually addressed, can have hundreds of variations within the same booklet by region or by market.

Four key elements make up PLA — InsertMinder (IM), CellBuilder (CB), PrintMinder (PM) and the PLA DATABASE (PLA/DB). In addition to the four main user elements this package also includes an archive element that acts as a graphic arts style Hierarchical Storage Management (GA/HSM) system and a workload balanced scheduling system (WBS) that allows users to both determine the statistical work capacity of a prepress operation and the appropriate staffing level and type to be employed given a volume and mix of work.

The application suite is built as a client-server system using either Apple Macintosh (preferred), Microsoft Windows 95 or NT Workstation systems as clients and NT Server systems as the server element. The database schema is written as ANSI SQL using no SQL implementation dependent verbs or structures. The current installation is based upon Microsoft SQLServer and employs Microsoft ODBC as middleware.

PLA employs a structure of workflow that we call "layered workflow theory". By this we mean that within the manufacturing plant there are multiple, exclusive layers within which different types of tools are needed for the best results.

The outer layer (Layer 0) consists of external data streams such as customer data, MIS data, and data from other non-prepress foreign processes (see Figure 1). These streams load jobs — data and metadata (data about data) — into the PLA/DB .

Within the next layer (Layer 1) are data generated within the prepress in commercial applications such as Quark, PageMaker, Illustrator and the like. Such applications are launched from within IM with data loaded from the database and, with work completion, the modified data are saved back through the database. Layer 1 data, while entirely controlled by the database, are considered unreliable in their application file form. Note that PLA time, date, data type, job number and user stamps such transactions.

Layer 2 data are reliable data. These data are derived from Layer 1 or Layer 0 data. Layer 0 data are preflighted for both inventory and PostScript content and "correctness". For PostScript files preflight includes parsing the PostScript file for type usage, distilling the file for RIP sanity proof (even though the PDF file is not used within standard production yet) and placing the derived EPSF into the "reliable and ready" area of PLA/DB. Layer 1 data are subjected to the same preflight and stored with the database as EPSF. Note that Layer 1 data are also subjected to one or more proofing rounds as cells (Figure 1). One preflight element that is not currently available but that is planned is the dynamic analysis of line width data which would allow the user to prevent non-printable line elements (e.g. small point size).

Layer 3 includes assembly of reliable components, by use of the CB application, into the page with page proofing. Imposition metadata are selected and validated at this layer as well.

Layer 4 consists of the imposition and printing process and those control systems involved at that level.

Tracking and Control: Insertminder

Think of IM as a data control center for the electronic job bag. IM controls access to PLA/DB, update and deletion of data within the database and is the access point through which one must go to reach any other application (external

Copyright 1998, IS&T

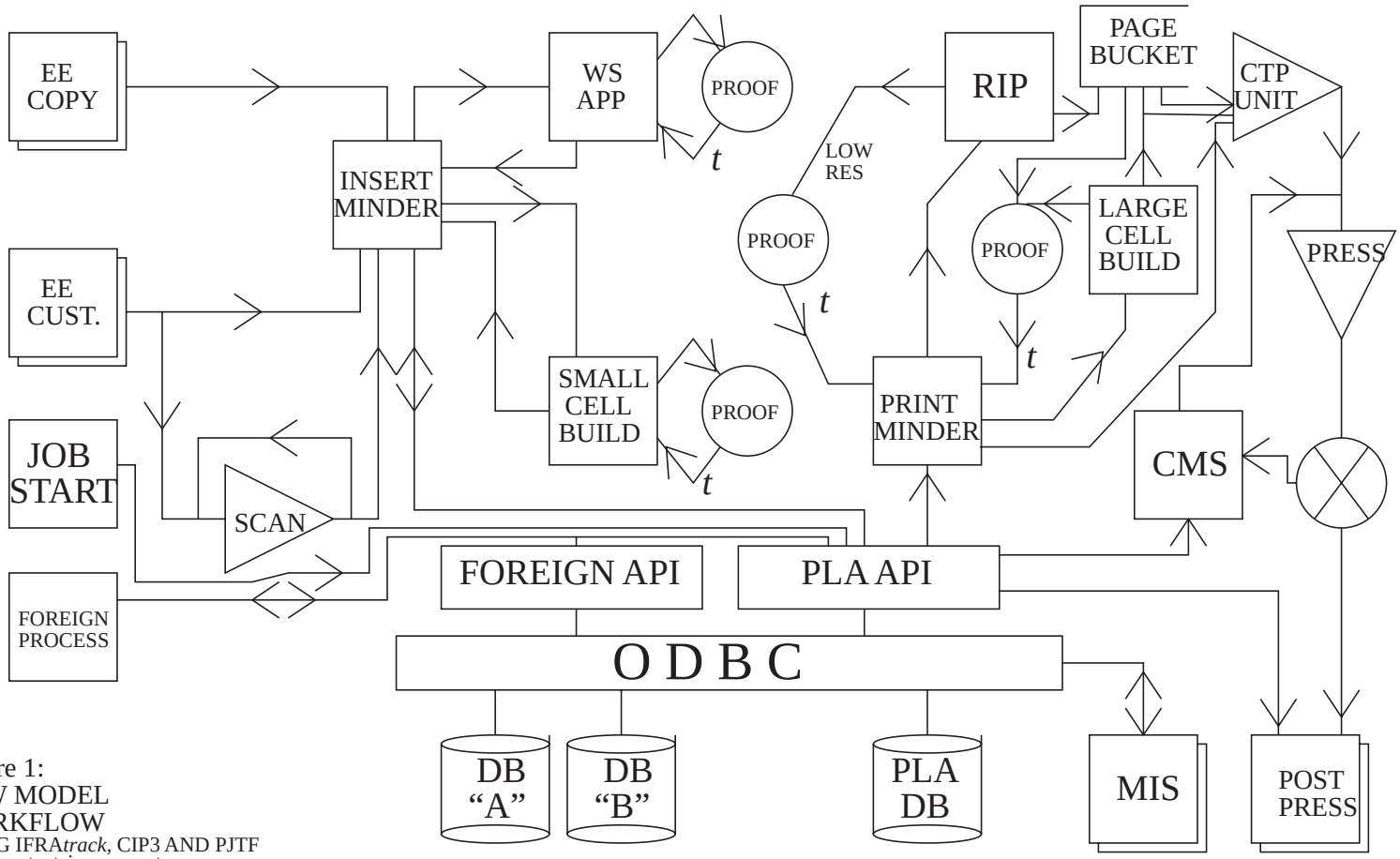

Figure 1:
NEW MODEL
WORKFLOW
USING IFRA*track*, CIP3 AND PJTF
t= trigger event

to PLA, such as Quark, or internal to PLA) that operates upon data housed within PLA/DB.

IM, as illustrated in Figure 2, provides numerous user definable views of PLA/DB. Structure and access level to the DB are limited by the user type and function. An IM view can be organized in such a way as to view by work and by priority — e.g. when a workstation operator logs on the system that individual can be presented with an IM view that is limited to the work that the operator is to do and the order in which they are to do it.

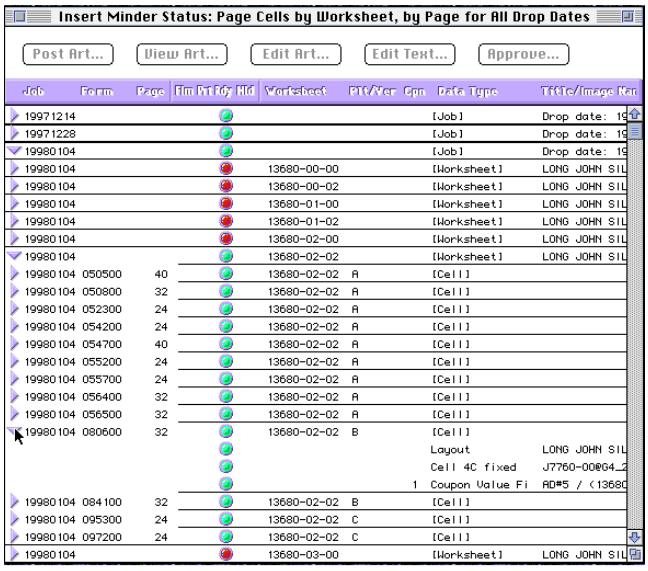

Figure 2. View of InsertMinder showing hierarchical file structure

Keep in mind that no data file can be opened within the workflow without going through IM. So, if an operator is to perform, say, cut outs on images in PhotoShop, then the highest priority data file is chosen in the IM view of that operator. This launches the file into PhotoShop and the work is performed. Upon closure the file is automatically saved into the correct referenced position in the data structure (more on these data structures in the database section).

By requiring all such movements to be performed through the standard IM launcher we are able to collect both open and close time stamps by operator, file and operation. This has significant impact upon the scheduling module, as described later.

Heterogeneous Element Construction: Cellbuilder

CellBuilder (CB), illustrated in Figure 3, is a new class of application that resides between the geometry tool and the imposition tool. This class of application has the characteristic of being similar to the geometry tool but does not operate at the level of detail geometry. The application also has the characteristic of being like an imposition tool in the sense that it positions elements within the larger geometry of the page or flat but, while it can act as a flat imposition tool, it can also act as a page assembly tool. Further, consider that this tool can deal with, display and manipulate positioning of multiple dissimilar elements in either traditional PostScript workflow intermediate file formats (e.g. EPSF, TIFF and the like) or in more nontraditional intermediate file formats such as PDF or TIFF/IT-P1 — each format being an element of a page, flat or whatever delineated printing element is required.

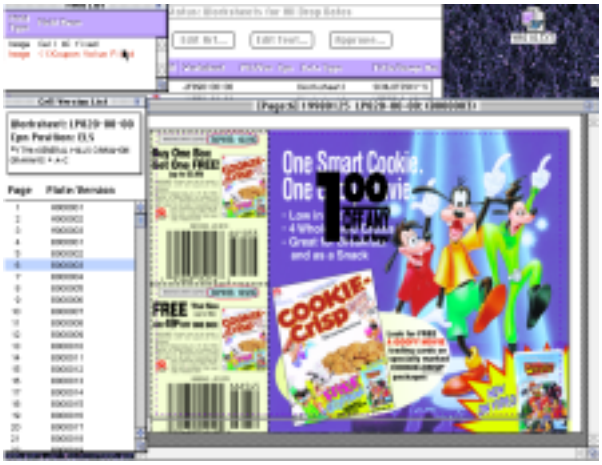

Figure 3. View of CellBuilder showing example coupon placement (the $1.00 off element). Note the left hand application elements of the example (Field Type and Field Name) and the variable plates in Cell Version List. In this case the cell graphic elements contain a copydot four color element and a contone TIFF (the $1.00 off coupon value field) generated by CellCutter.

For the sake of clarity, Figure 1 shows two CB modules (LARGE and SMALL). These are identical applications used at different points of the workflow. SMALL CB operates, nominally, on the page metaphor allowing simple assembly of pages from previously validated elements that are available in the PLA/DB. Such assembly includes variable field elements placed by a sample exemplar, defined at the IM level and printed by rule at the PM level.

Why create a new class of tool? First, there needs to be a clear separation between the "creative" element of the workflow and the production element of the workflow. In other words, we need to allow the workflow to operate on a natural level of granularity. Creatives deal with the detail content elements while production deals with completed cell data. Cell operations represent multiple tools with multiple workflow element details and each workflow is different both intrinsically and by job type. Second, significant amounts of production require that prepress "splice" together dissimilar elements created in different applications or from remote and non-contiguous workgroups.

By observation, in prepress production environments we have found that designer tools, such as Quark, while extremely useful to the creatives, are accidents waiting to happen in production. These traditional geometry tools are

unintegrated, point of action tools with trivial file systems-for ease of use by artists. These things simply don't scale (it's like trying to scale a Geo Metro into an Abrams tank) and require pathetic scripting and file control efforts to even begin to bring order from chaos.

Keep in mind that the eventual object of CB is to deal with previously validated application data that has been converted into the digital equivalent of film. So, as elements are received by prepress they are inventoried, classified, loaded into the DB, preflighted and proofed.

Classification determines the particular type of workflow the cell will be sent to.

Figure (7), drawn from the Valassis design study, illustrates that there are only three possible digital prepress workflows — Art Supplied, File Supplied and Film Supplied. So, data that arrive as geometry and image elements (Art Supplied — the classic PostScript workflow) are processed within that workflow to the point of PostScript or distillation — which yields reliable cell data. Data that arrive as, say PDF or TIFF/IT are processed as file supplied data and copydot data are processed as Film Supplied. Everything is reduced to a common, reliable set of pre-validated file elements.

CB, like any other application within PLA or used by PLA, is launched from IM.

Smart, Variable Field Capable, Printing Control: Printminder

One of the implications of the printing industries move to computer to plate systems is the absolute need for validation of data *prior* to platesetting. The purpose of CB and IM is to provide such validation. The purpose of PM is to execute — seemlessly — the instructions that prepress provides to PLA to manufacture the particular job. Indeed, all steps prior to PM are merely either data acquisition or metadata building because PM actually produces, by way of the platesetter or imagesetter, the job in its' complete form.

PM is automatically invoked from within the IM process by either completion of an image unit (flat, Miso page or whatever is defined as the printing unit) or through a timed demand. PM CANNOT be invoked (without manual override) without all known elements of the print unit being complete.

Further, the PLA/DB retains a central database of type for both PC and Macintosh systems. Typefaces are linked to the given cell element to be printed and, if the typeface is not available to the server then the cell, and consequently the image unit, CANNOT be printed until the condition is corrected.

For those users who build cell elements locally to the prepress workflow, PLA provides a type capture function. Keep in mind that all applications are launched from IM. As PostScript is written from the application to the PLA/DB it is parsed for type information and PLA/DB stored faces are associated with the cell. PLA reaches out to the workstation and captures typefaces present in the data being stored but missing from the PLA/DB.

An important element of PM is the ability to automate variable field printing, press versionalization and load balancing within a set of imagesetters or platesetters.

Some workflows require very significant variations in either type or image within a given press run. In the case of Valassis, as an example, black plate coupon data may vary within the 200 or so markets that the coupon booklet is targeted to. Such variations are fixed field changes in legal data and barcode or are free form type changes that occur within a given spatial coordinate set. Such data are a significant problem in management for the prepress as many of the changes are either fairly subtle or not human readable. PM allows the automatic production of such variations with a "print by example" technique. A single master setup is done in CB and each variation is correlated and printed automatically (see Figure 4).

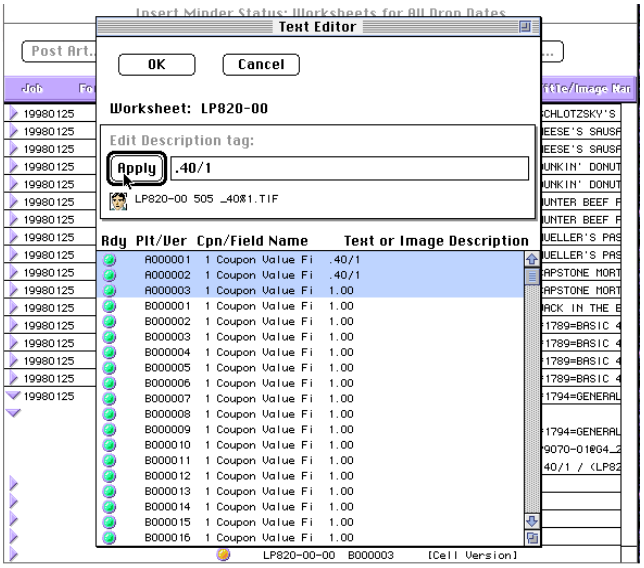

Figure 4. Assignment of text variable field data to ganged variable plate output.

Large production operations also have the problem of multiple presses that may be printing the same job or job version. These presses may also be remote to the prepress. PM allows the user to specify multiple impositions and multiple press fingerprints for automatic, on the fly production.

In other circumstances multiple plate sets need to be built for one or more presses for rapid plate change for multiple versions. In the case of multiple black plates being needed for a single CMY set with multiple platesetters being used, the same platesetter should be used for the K variants as was being used for the CMY (there are several reports of commonly used plate devices having "unique" measurements yielding lack of fit for otherwise orthogonal data).

In the initial installation at Valassis pinned page film output was specified. A Purup Maestro was modified to provide two film "channels" — each with pins — however, each channel was physically unique due to unavoidable mechanical circumstances. Thus, we were presented with an asymmetric queuing problem as K plates could not be done on a different channel than the CMY. However, the print

load needed to be balanced to allow as continuous an output as possible. For circumstances such as this, PM uses queuing theory to build the initial submission of data to the print channels and prevents odd plate reruns or variant K plate data from being printed by the wrong channel. Channel balancing is based upon a "look ahead" algorithm which tries to keep the data stream constant to both sides of the drum.

One thing to note with the PM process in Figure 1 is the "page pump" approach to imaging. This logical device allows the user to achieve many of the benefits of post RIP imposition prior to RIPing by dealing with flat data on a page level — divorced from the flat fold and cut marking data. Thus, potential page level changes can be made and, as PM controls its' own imposition system (CB) the process is automated and transparent to the user.

Acquisition and Control of Data: PLA/DB

PLA/DB is a large, ANSI SQL repository of job metadata that points to high resolution data, geometry files and variable field data. There are several elements of PLA/DB that should be noted:

Command hierarchical file structure: PLA/DB builds a standard hierarchical file structure for each job and job element of the pointed to variable length data. File structures are universally standard and, in case of disaster, can be accessed manually.

Extensive hierarchical file structure: PLA/DB automatically spawns duplicates of its structure across logical drives as jobs outgrow a given physical drives capacity.

Extended image database: PLA/DB, by its' very nature is an image database. The design logic of PLA is such that all data files are treated as image objects so, as a consequence, all data files are available through an image database metaphor.

File and image version control system: PLA/DB draws upon the UNIX development metaphor and provides the graphic arts user with both a revision history and, if desired, the revision history is tied to the actual versionalized data object. This technique is used for image data, geometry data and PostScript/PDF objects.

Job Work History:

When used with the job tracking element of statistical wokload and capacity planning (WBS), the database collects user time on task, task type and task time stamp in all applications launched from IM — automatically and without user input.

Data Driven Structure:

PLA/DB, while a classically structured SQL design, is a framework database. That is, the database use is defined by the data environment that PLA is to be used in. This allows great flexibility within the product but does require a fairly extensive definition of data upon the part of the user — something that should be done anyway.

Loose Coupling:

All PLA applications stand alone upon PLA/DB. All communication between application types and workflow steps pass through the database and are, then, by nature, asynchronous. This allows for time buffering to take place within the workflow making it more flexible to the variables within the actual shop flow conditions.

Intrinsic Parallelism:

As noted earlier, process driven workflows have difficulty dealing with parallel and asymmetric tasks within a workflow. Further, due to the tight coupling assumed within process driven workflows, resource utilization is reduced as the scale of the integration increases. Data driven workflows are intrinsically "loose" and accommodate parallel process, ad hoc processes and feedback process delay while providing to the workstation user (and the system in general) as much resource utilization as possible. Note that the data driven model is also of a more finely granular nature (being cell based rather than page based) and allows the user to work on what is available at the cell level.

Cell Cutter: CC

The PLA CellCutter (CC) module was designed to allow users to rapidly and cleanly cut image data from ganged scans and to trivially assign these data to the database. Figure 5 illustrates this process. Note, that within the variable field printing environment, one can fill multiple cells or forms in by a single posting.

This system, while used on K plate data for the most part now, is fully color capable.

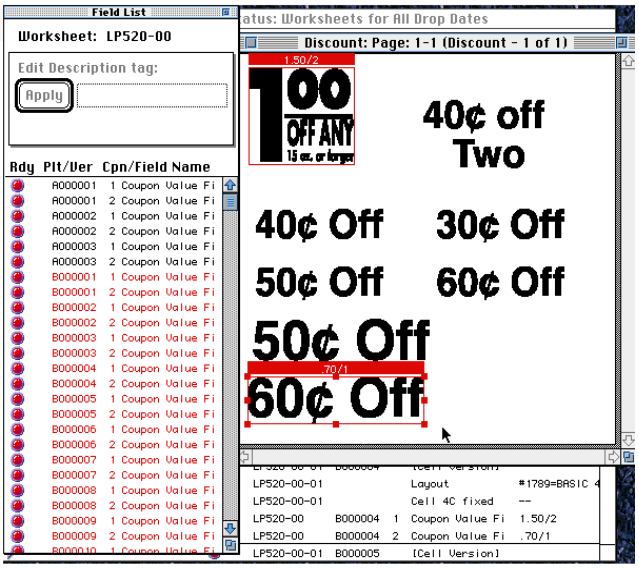

Figure 5. Cell cutter assignment of variable field plate output.

Graphics Arts HSM: GA/HSM

Hierarchical storage management (HSM) is a well known technique used within the business information technology (IT) environment. These systems track file

usage. As files age they are automatically moved from online storage (fast access) to nearline (slower access) storage. Over time and as the files age, unused files will fall into a classification that will allow the system to automatically remove them from direct access to either tape or some other offline storage.

HSM's can be quite sophisticated and some even employ rule based systems to filter files through the system at different rates based upon data content. However, such IT HSM systems have intrinsically limited value to the graphic arts user. For one thing, graphic arts file sizes are considerably larger than those of the typical IT system and usage criteria may not be strictly based upon an aging criteria.

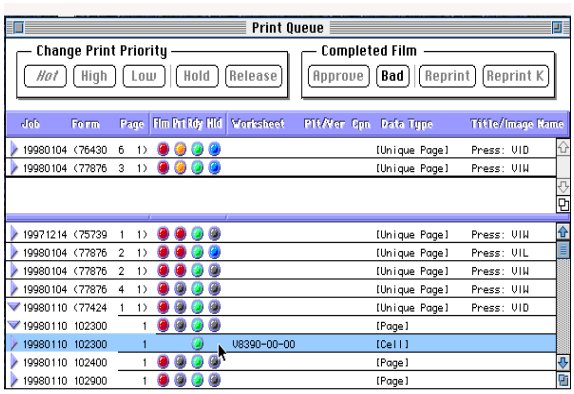

Figure 6. Print Queue showing active and complete print processes and respective process status. Note that printed elements are either marked "approved" or "bad" based upon either physical inspection or known process failure.

PLA employs a set of HSM file control logic that is based upon the graphic arts users needs. The GA/HSM is event driven rather than aging driven. So, a catalog that is built with a know periodicity will be moved offline at print approval (see Figure 6 for the approval process) and will automatically be brought back online at a user defined period prior to reuse need. The command logic can either be date driven or by user demand.

Further, the GA/HSM eliminates the concept of nearline storage. Magnetic optical or CD-ROM juke box systems are very unreliable under the best of circumstances and are simply accidents waiting to happen. We view online as being hardware driven RAID while our nearline metaphor is the software RAID using the new and very cheap 43 Gbyte drives.

Offline storage is via DLT or, better, ALT tape stacks. These units, particularly the ALT, are very fast and are highly reliable.

Statistical Workload and Capacity Planning: WBS

No one in the prepress world really knows what their capacity is or what their throughput can be. The variables within prepress workflow were, here-to-fore, too complex to track in detail. With PLA we have an application that provides the level of data that are required for capacity planning, work scheduling and reasonable throughput estimation.

PLA gathers massive amounts of data about the one production element that is generally very poorly tracked — labor. The very nature of user interaction with PLA, as noted earlier, captures workstation operator actions by logon, date, time of file checkout, workstation action, and time of file storage.

The WBS module, while still very primitive, allows the user to ask some interesting questions. For instance, what workstation user combination is best employed to solve a give job type or, with measured machine data, what is the actual throughput of a given workflow.

Further, the actual labor cost of a job can be tracked in detail, trivially and without anyone within the production organization having to fill in forms.

Eventually, we plan to employ certain elements of operations research which will allow very tightly scheduled (for prepress) workflows.

Conclusion

Production system wide automations (CIM's) are required for large scale variable field printing if for no other reason than the shear complexity of data change within heterogenously derived source documents. PLA operates by exemplar placement of field data which allows totally automated field substitution of data within such complex origin documents.

Current implementations of PLA are restricted by the available output transducers. Web offset presses simply cannot do directly targeted documents other than by the fairly crude inline inkjet approaches. Color electrostatic printing technologies, while easily addressable by PLA, simply do not offer either high enough volume production or appropriate cost per folio to make wide scale use of such transducers economically attractive.

On the other hand, the production of four color shells via offset printing and the subsequent electrostatic overprinting of such shells with variable black data is both quite possible and done in fairly large numbers daily by PLA.

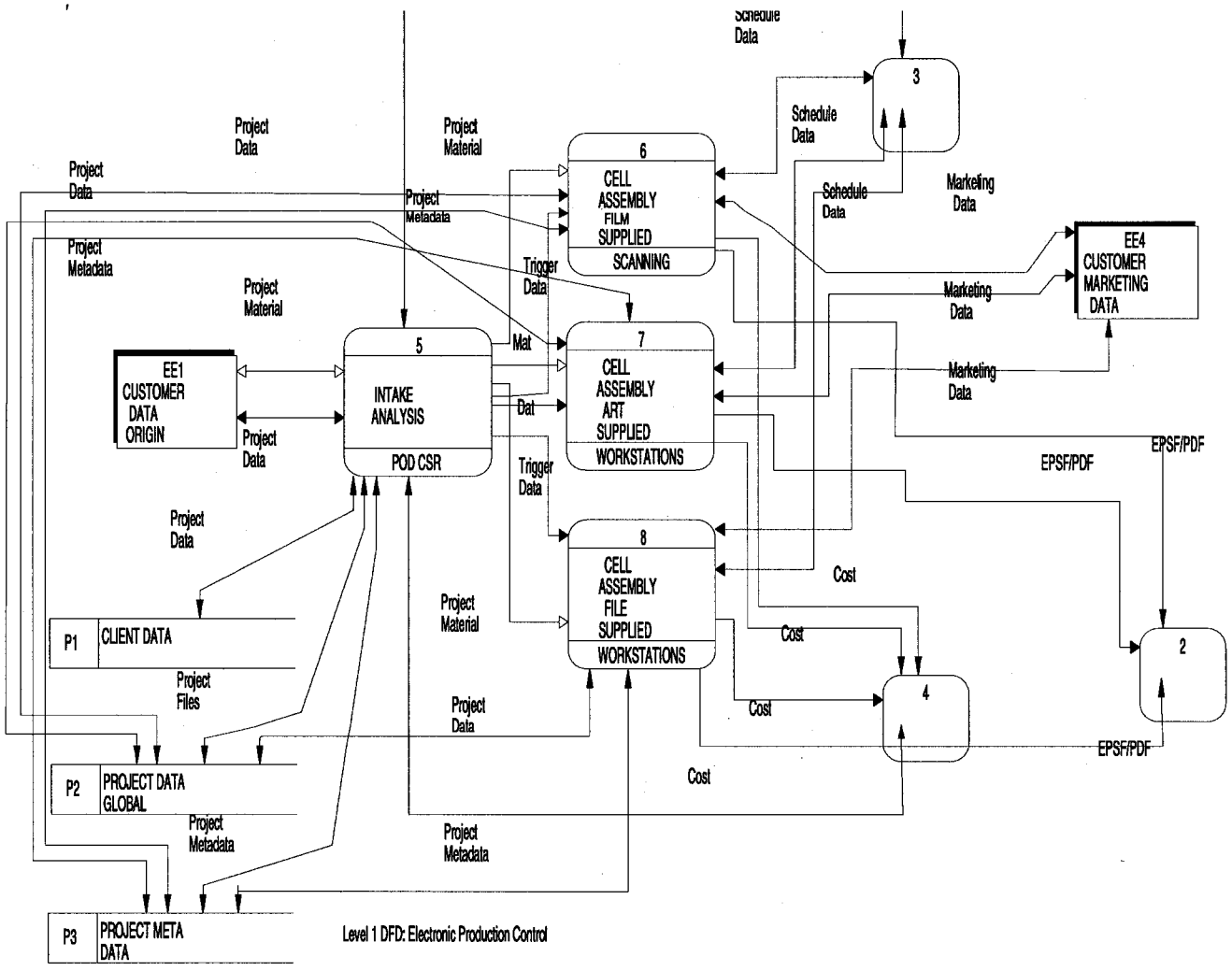

Figure 7: Section of prepress system model