3 Workflow Models

Abstract *Chapter 3 is about models that are used in print production. First, flowcharts are briefly discussed. After that, the most common model for print production is explained in detail: the process-resource model. This model is the basis of the Job Definition Format (JDF) that is the topic of Chapter 4.*

Keywords *Production Model, Process-Resource Model, Flowchart, Activity List.*

Since automation eliminates the ability to make ad-hoc decisions during the production of a print product, it goes without saying that it requires rules on how a type of product is made. While rules are always needed, even without production automation, the difference between non-automated and automated production is that for the latter the rules must be formally defined. It is not enough for the employees to have these rules in their heads. Moreover, these rules should not always be created anew, though perhaps differently for different product types. The rules should be designed according to a uniform scheme – in other words, based on a model.

Of course, this does not imply that the manufacturing process of a print product must always follow these rules no matter what, or even that a production that has been started cannot be changed or stopped. And since order changes are quite common even after the production has started, or for technical reasons, it must also be possible to change the production rules for a specific order. In addition, it must be possible to intervene manually if individual process parameters do not correspond to the product's specifications. Furthermore, automation may be questionable in an environment with more one-off print jobs than standard print jobs.

As usual, there are different options for modeling a production. Let us look at three of them.

3.1 Activity List, Process Chart

Perhaps the first thing that comes to everyone's mind is to simply list the required production processes, i.e., to create an *activity list* or to-do list. That is fairly easy and great for brainstorming, but not so good for a structured model. Obviously, there is no order specifying which process should be executed first, which one is next, and so on.

An ordered list or a process chart can do the trick. These are actually quite common and sufficient for the first draft of a workflow description. A process chart is even better because it also allows

T. Hoffmann-Walbeck, *Workflow Automation*, SpringerBriefs in Applied Sciences and Technology, https://doi.org/10.1007/978-3-030-84782-1_3

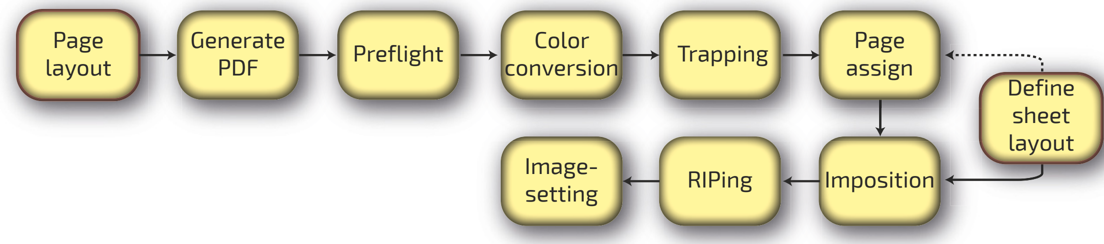

Figure 3.1: Process chart for offset prepress

branching and multiple starting points. Figure 3.1 shows such a chart for a typical process sequence in the prepress phase of offset printing. In this figure, I omitted the creation of layout elements such as text, images, and graphics. There are two starting points: "Page layout" and "Define sheet layout".

Process charts are still too vague, however. Neither the prerequisites (inputs) nor the outcome (outputs) of a process are defined. For example, what is the outcome of the "page assign" process? Is it

- a list of PDF files and page numbers,
- a link to a PDL file with pages in the reading order,
- or an assignment of pages to the position of the sheet layout templates?

Definition

PDL stands for **Page Description Language**. PDF is certainly the best known PDL.

What kind of information is needed for defining a sheet layout? What is it that the imposition process and the page assignment process actually do? What kind of data is needed for image setting? Obviously, an automated production system requires unambiguously defined processes as well their inputs and outputs.

The chart shows only an "ideal" workflow. However, every process has at least two status outcomes: success or failure. Figure 3.1 contains neither proofs nor reviews, but they could easily be added.

3.2 Flowchart

Flowcharts are well-known and widely applied. They are composed of symbols and are used commonly in data processing. They serve primarily the visualization of a program designs during software development. However, they are also used in more general models of business and production processes. They became a DIN standard way back in 1966 (DIN 6001, 1966) and were adopted as an ISO standard in 1985 (ISO 1985).

Figure 3.2 shows an example. The blue graphical elements symbolize the start and the end. The green diamond-shaped symbol represents a *decision*. The question allows only two mutually exclusive answers, *yes* or *no* (alternatively *true* or *false*). The red element describes a *predefined process*, which is ideally again described by a separate flowchart. In addition to the three graphical symbols presented, there are of course a few more. A circle represents a *connector* (see Figure 3.3), a simple rectangle indicates a process or operation, and so forth. Please note that only the shapes of the elements are specified in the standard, not the colors.

Figure 3.2 demonstrates a typical communication exchange between PP and PB, in case the PP generates a proof.

In contrast to the activity diagram of the last section, in flowcharts *decisions* are part of the concept. In this respect,

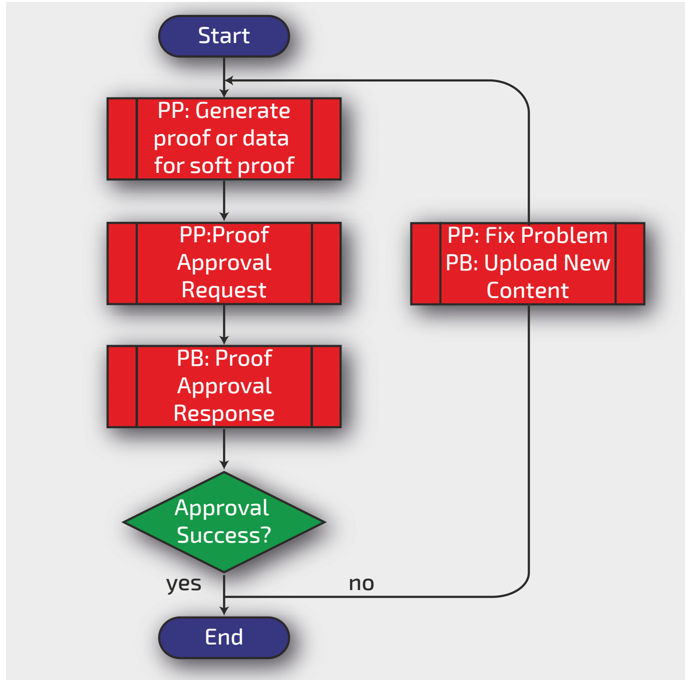

Figure 3.2: Proof cycle between print provider (PP) and print buyer (PB). Source: PrintTalk 1.5, Figure 3

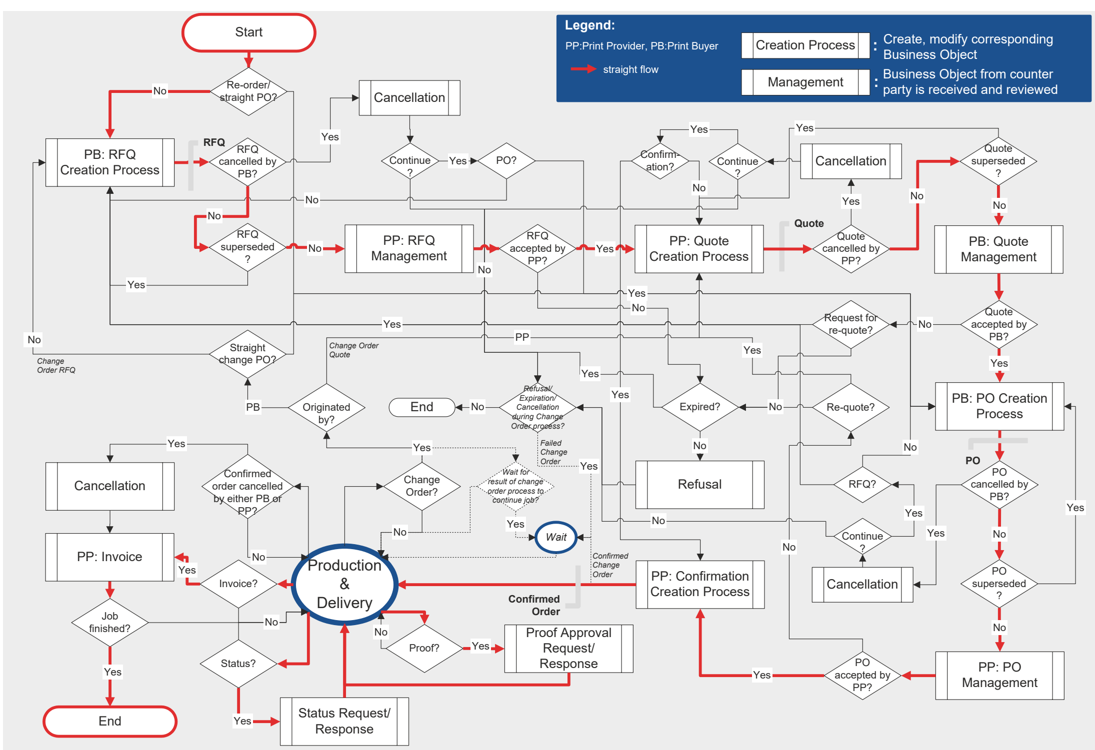

Figure 3.3: PrintTalk workflow diagram. Source: PrintTalk 1.5, Figure 6 , © 2000-2020 CIP4

flowcharts are particularly suitable for modeling flows that take different process states into account. Figure 3.3 displays *decisions* between *predefined processes*. Even though it might be important to consider all possible branches in a workflow, things can quickly get very complicated and confusing if an entire production model is supposed to be specified. Thus, flowcharts might be better for specific parts of a workflow than for the interaction of large numbers of processes.

The decisive factor, however, is that flowcharts usually do not define the inputs or outputs of processes. This leads us to the last and most important model for the graphic arts industry in the following section.

3.3 Process-Resource Model

Defining preconditions and the outcome of processes is the strength of the *process-resource model*. Here, every process has input and output resources. Figure 3.4 shows an abstract example. The process 1 has three input resources on the left-hand side and one output resource on the right-hand side.

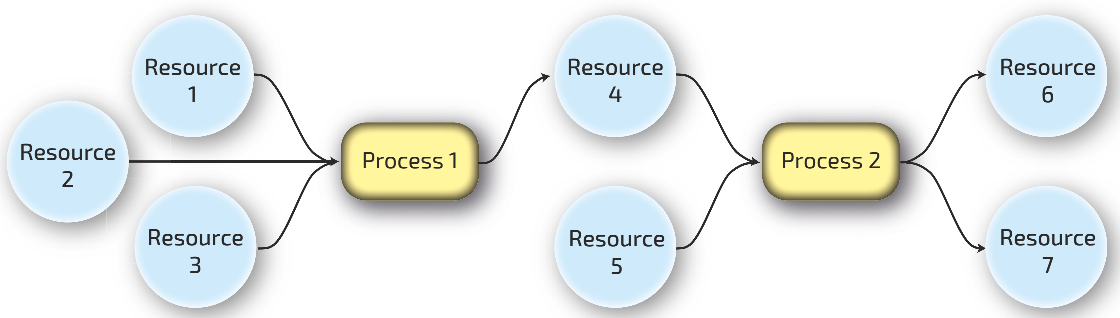

Figure 3.4: Abstract example of a process-resource model

The yellow rectangles represent processes; the blue circles are resources. We already discussed these two terms in Section 1.1. A slightly inaccurate but easy take-home message is that processes are verbs and resources are things. A process normally has several input resources, i.e. resources that are needed for the process to execute. Moreover, the process generates one or more resources, which are called output resources. Processes and resources always alternate. A process can only generate a resource, not another process. The similar statement holds for resources: A resource can only be input of a process, not of another resource.

Please note that resource 4 in Figure 3.4 is simultaneously an input and output resource, while the others are input or output resources only. Resources 4 is the most important one because it determines in which order the processes must be executed. In this text, I will call such resources “transitional”. In the following, I will concentrate mainly on transitional resources in order to simplify our charts. That is, I will omit quite a few input resources in the diagrams.

The process-resource model is the basis for workflow automation in the printing industry. The basic idea is this: A process may start automatically if all input resources are available.

Digression: Let's make pancakes

Please feel free to skip this digression. It is not essential for the understanding of the following but only supposed to explain the model with the help of a somewhat odd example: making pancake. It proves, however, that a process-resource model is useful for all kinds of projects.

Figure 3.5 shows four processes: *Mix*, *Fry*, *Separate,* and *Beat*. Each process has several input resources. The mixing process, for examples, has:

- Consumable resources, i.e. ingredients, such as sugar and flour
- Tools, such as a bowl and a stirrer
- Operators, such as the chef

Four resources are transitional. They are marked with a red border. The resource *Pancake* is the final product and hence an output resource only.

This example of making pancakes is not complete. Some details are missing. In particular, I did not…

- …define details of the end product, i.e. the number and diameter of the pancakes.
- …specify any quantities, for example, the number of eggs.
- …describe the tools, for example, the diameter of the bowl.
- …specify the kind of class a resource belongs to; for example, the *chef* is an operator and not an ingredient.
- …specify the production times. Obviously, a production end for a process needs to be observed in order to meet the delivery date of the final product. On the other hand, a process should not start too early, because of
  (i) last-minute changes that might be required and
  (ii) the output resource of the process might become unusable or damaged if it is stored for too long. Moreover, storing items increases costs.
- …identify details of the processes, for example, how long they should execute (such as mixing or beating).

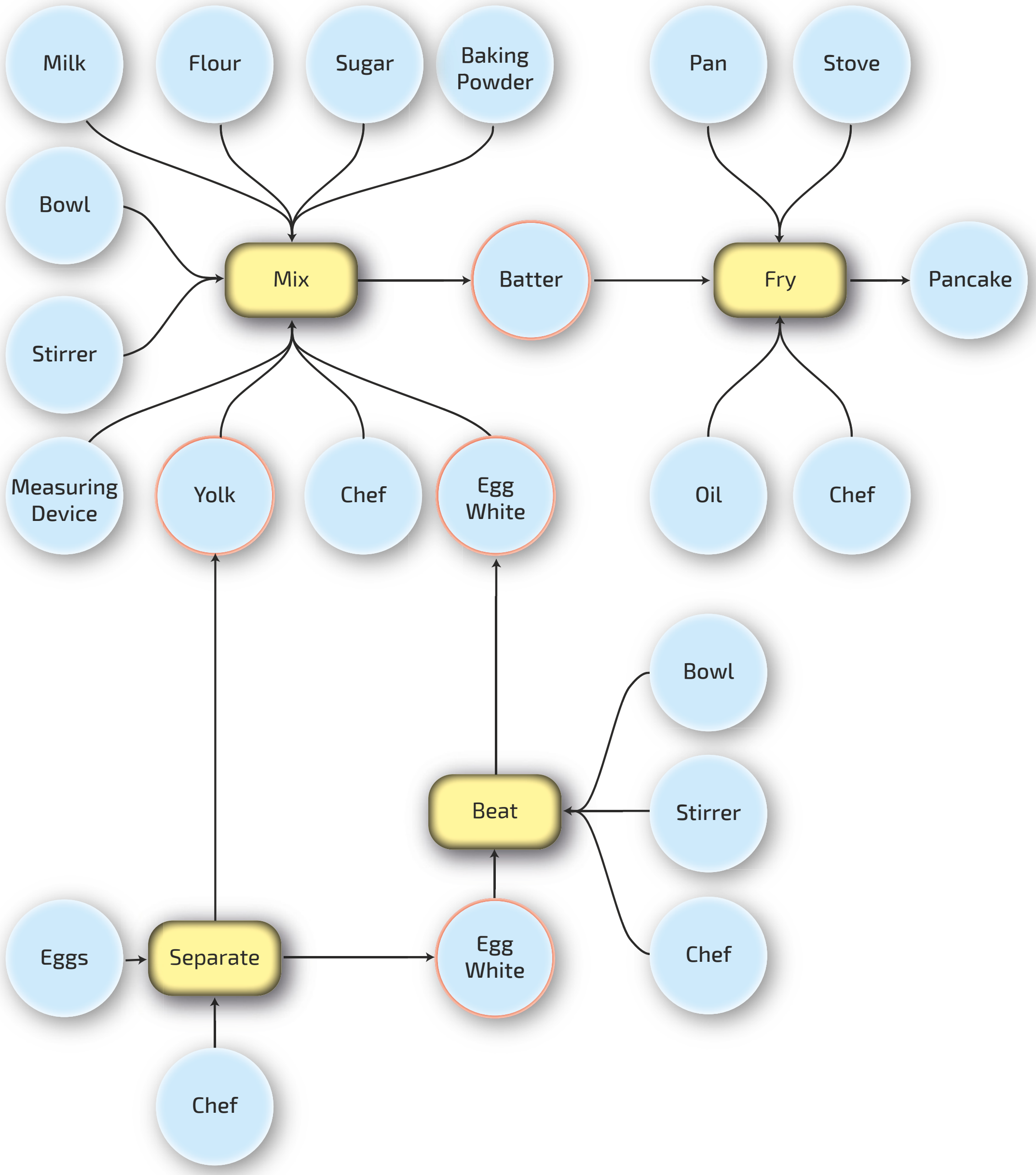

Figure 3.5: Process-resource model for making pancakes

The last point means that usually each process needs one more *parameter resources* as input, which describe further details.

In the context of automation, we can replace the chef by some robots or devices. That is, one robot for each process. To ensure such a production, each process and each resource would have to be specified much more carefully. Moreover, the goal would be to automate cooking in general, not just the production of pancakes. Thus, we need to specify all processes and resources for cooking.

Moreover, robots need to understand this be able to execute the processes automatically. They should be able to communicate with each other and/or a central controller. The specification of the processes and resources should be based on an open standard so that the bakery can buy robots/devices from different vendors – the mixing robot from vendor X, the frying robot from vendor Y, etc.

Let us assume that there is a specification and that there are vendors who manufacture such robots. We still need someone who prepares and integrates them into a production line, plans and reacts to new requirements (for example, when a new recipe is introduced), and intervenes if something goes wrong. This is a task for a workflow manager in a fully automated baking plant. In addition, a workflow manager must think through the entire process chain from start to finish in advance. Manufacturers mainly consider the production segment of their own workflow solutions only.

End of Digression

Normally, a process relates to many input resources and only to one or two output resources. Figure 3.6 shows the model of conventional printing using plates such as sheet-fed offset. This example comes very close to the more general model of CIP4. However, as already mentioned in the preface, I will not use the abstract and general resource names of CIP4 in Chapter 3 but rather the terms commonly used in the graphic arts industry.

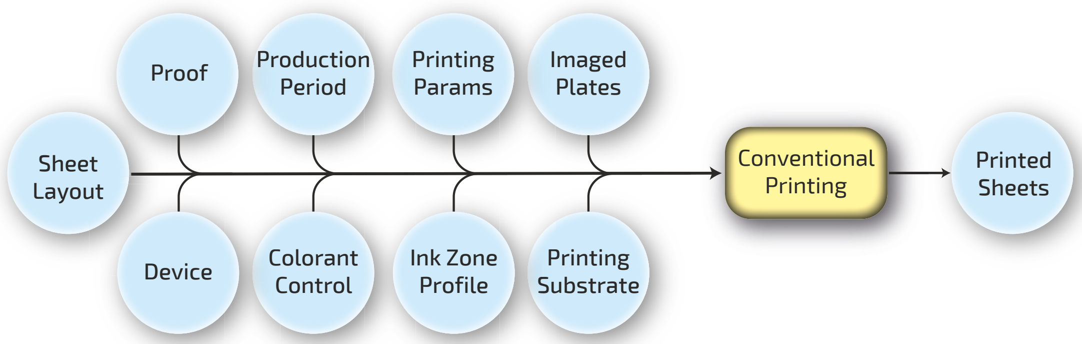

Figure 3.6: Process-resource model for printing process

Still, the resources in Figure 3.6 might need some further explanation:

- Imaged Plates: In lieu of plates there could be cylinders or sleeves, above all for gravure or flexographic printing.
- Printing Params: Device setup information – for instance if the fountain solution modules in an offset press or the heating modules should be turned on or off.

- Production Period: Time interval in which the printing process should be executed.
- Proof: Page or imposition proof produced by the prepress department.
- Sheet Layout: Types and positions of control elements on the sheets – for example, for registration or color control.
- Device: ID of the press and its capabilities – for example, its maximal run speed.
- Colorant Control: Order of the inks.
- Ink Zone Profile: Information about the amount of ink that is needed along the printing cylinder for offset presses with traditional inking units (no anilox rollers) for each separation.
- Printing Substrate: Details about the paper or foil, such as thickness, type, and size.

Figure 3.6 does not show all possible input resources for the process. For example, in flexographic printing the mounting tape should be specified as well. Input resources are mostly non-mandatory. Ink zone profiles, for example, make no sense for an anilox offset press.

Video about Workflow Puzzle

The last figure shows many input resources for a single process. For the initial planning of the production workflow of a specific product type, the processes and the transitional resources between the processes are of primary importance, as already mentioned. To add all input resources at the very beginning might become overwhelming.

Richard Adams and I have written a little tool in JavaScript that applies the process-resource model to different situations in print production. We called it *Workflow Puzzle* (Hoffmann-Walbeck and

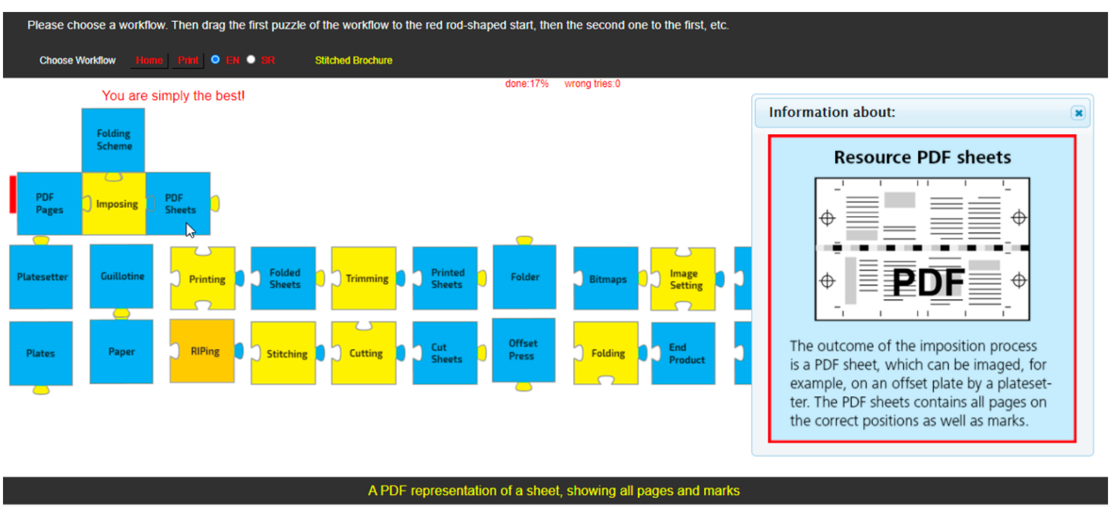

Figure 3.7: Interactive tool for creating a process-resource model

Adams, 2021a). Please try it out yourself. You will find it at (Hoffmann-Walbeck and Adams, 2021b)[1].

In this script, there are different workflows you can choose from, such as Stitched Brochure, Deck of Cards, RIPing, LAMS Flexographic Plates, etc. Figure 3.7 shows the workflow of a stitched brochure. As before, processes are yellow, resources are blue. Holes and bulges define the input and output relations between neighboring cards.

Figure 3.8: Processes and resources inside the process group RIPing

The idea is to drag these cards into the right order. The little red bar is the starting line. The workflow runs from left to right. If you pick the correct card, move it close to the predecessor card and release the button, it will snap into its position, and you will hear a sound. There is also text-based feedback. If you do not know one of these processes or resources, you can get information about it by double-clicking on the card. An information window such as the one on the right-hand side in Figure 3.7 will pop up. In addition, if you hover over a card, a short explanation appears at the bottom of the browser window. In the screenshot, there are four cards already placed on the correct position; all others are random. The software counts the wrong attempts. In my opinion, the best players are those who have the fewest false tries. Unlike with other online games, speed is not an issue.

Video about Workflow Arrows

In Figure 3.7, the *RIPing* card has a slightly different color compared to the process cards because, strictly speaking, RIPing is not a process but a process group. It includes several processes such as *Interpreting*, *Rendering* and *Screening*. Figure 3.8 shows the RIPing workflow in *Workflow Puzzle*.

By the way, at (Hoffmann-Walbeck and Adams, 2020a)[1] you will find a similar tool called *Workflow Arrows* (Hoffmann-Walbeck and Adams, 2020b) that allows you to define processes, process groups and resources yourself and connect them freely (see Figure 3.9).

In the first attempt, there is no need to distinguish that carefully between processes and process groups. However, when the model is refined, we often need to replace the process group with individual processes. This is true not only in graphical modeling on paper, but also in a real production environment. An MIS may not be able

Definition

Interpreting is the first process inside a RIP. PDL data like PDF is read, parsed, and simplified for the next process. For example, Bézier curves are iteratively approximated by polygons, images are decompressed etc.

Rendering converts graphical elements, which might include vector data, into contone images (byte maps).

Screening converts contone images to monochrome images (bitmaps).

1 This script is merely a prototype and not a professional program. We tested it only on a Windows PC and on a Macintosh. Especially on touchscreen devices it will not work well.

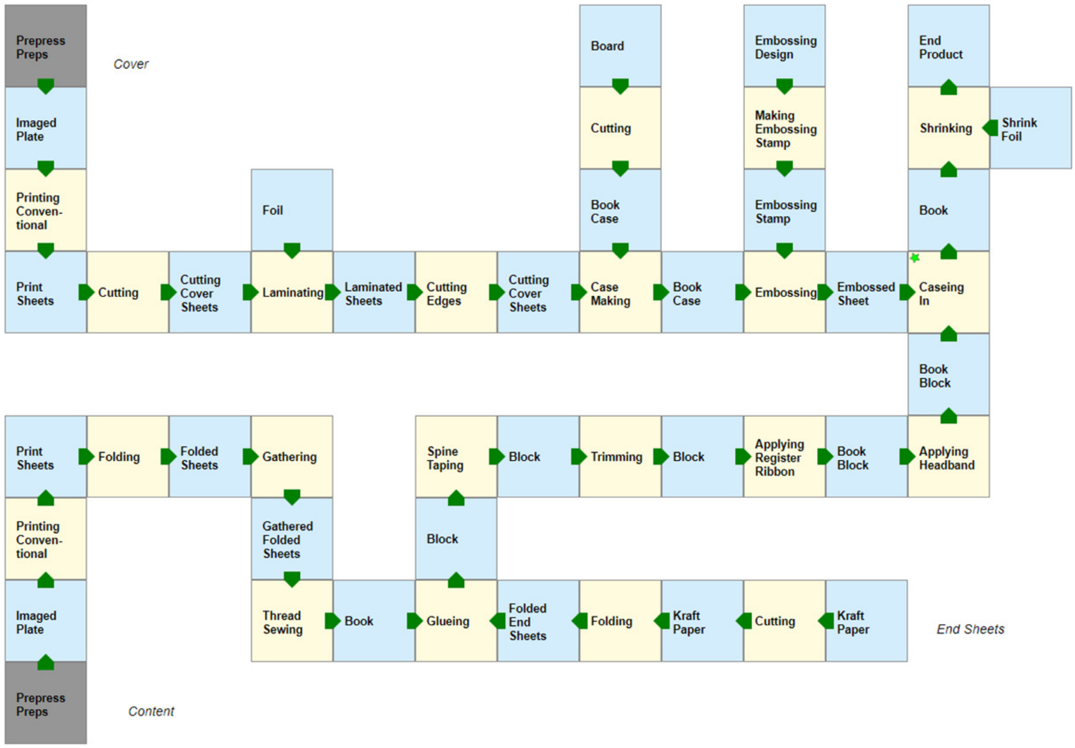

Figure 3.9: Process-resource model for book making

to specify processes individually but might only create process groups such as RIPing. A production controller for prepress that knows the important details will subsequently fill such a process group with individual processes. An MIS might even issue a very general process group such as *Plate Making* or *Prepress Preparation.*

Figure 3.10 provides a more complete picture of processes and transitional resources for a prepress workflow in commercial printing. It does not show the artwork creation processes but rather starts with PDF pages which a PB often sends to a PP.

Unfortunately, this visualization of a process-resource model can easily lead to a misunderstanding. The arrows do not imply that files are copied into the "input area" (such as a hot folder) of the next process. Often, the first process will store the output resource on a server and merely notify the next process where the data is located. In fact, passing information from one process directly to the next is also likely to be an exception. Usually, the process will only inform the MIS or the production controller where it has stored the output resource. The MIS or the production controller will then inform the next process about it at the given time without the original process noticing anything.

In Figure 3.10, the tiny red arrows symbolize this data exchange between devices that execute these processes with the management software. The black arrows show only the logical communication path, not the real one.

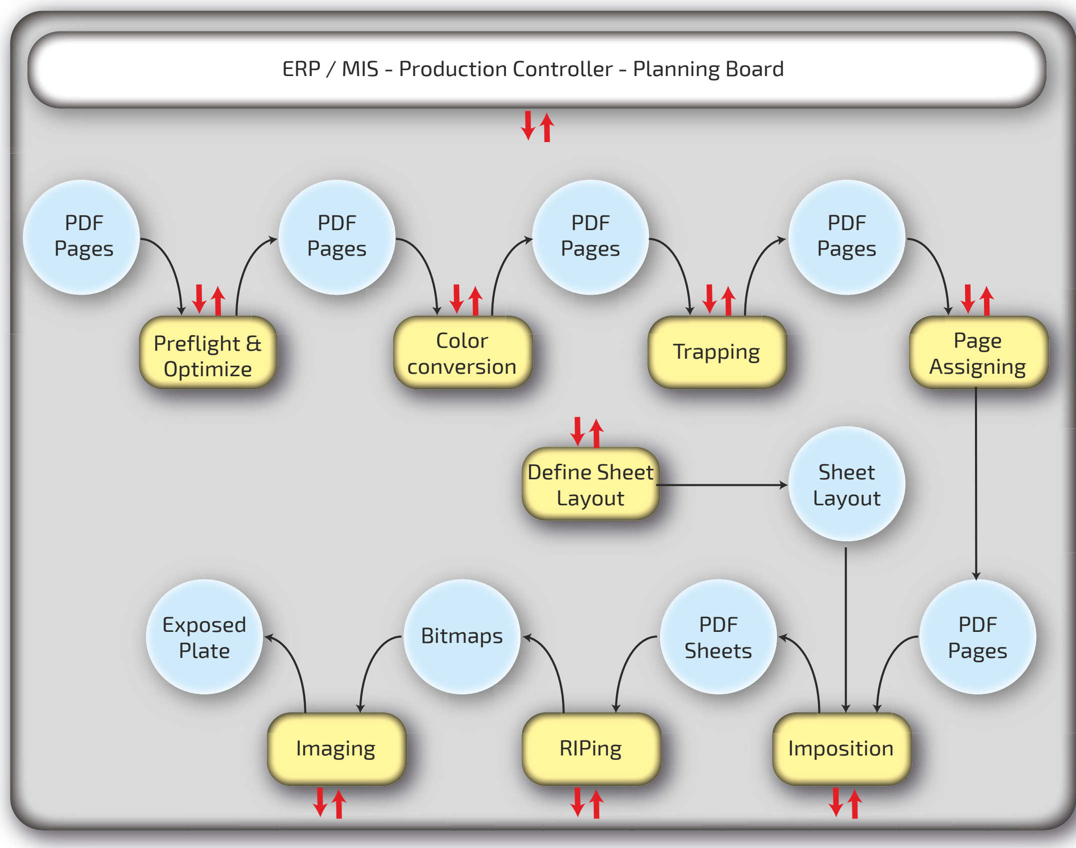

Figure 3.10: Process Resource Model for prepress in commercial printing

Reference

Hoffmann-Walbeck T., Adams R. (2020b): Visualization of the Process-Resource Model for Workflows, GRID 2020 Proceeding, p. 623-626. Available at: https://doi.org/10.24867/GRID-2020-p70 (Accessed: 15 June 2021).

Hoffmann-Walbeck T., Adams R. (2020a): JavaScript “Workflow Arrows”. Available at: https://www.ryerson.ca/~wdp/workflow-puzzle/print/Test/Workflow_Arrows.html (Accessed: 15 June 2021).

Hoffmann-Walbeck T., Adams R. (2021a): Online Workflow Puzzle, Journal of Graphic Engineering and Design, Vol. 12, No. 1, p. 5-9. Available at: https://www.grid.uns.ac.rs/jged/?pid=1022 (Accessed: 15 June 2021).

Hoffmann-Walbeck T., Adams R. (2021b): JavaScript "Workflow Puzzle", Available at: https://www.ryerson.ca/~wdp/workflow-puzzle/print/print.html (Accessed: 15 June 2021).

ISO 5807: 1985, Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts. Available at: https://www.iso.org/standard/11955.html (Accessed: 15 June 2021).