Supporting the BPM life-cycle with FileNet

Mariska Netjes, Hajo A. Reijers, Wil M.P. van der Aalst

Eindhoven University of Technology, Department of Technology Management,
PO Box 513, NL-5600 MB Eindhoven, The Netherlands
m.netjes@tm.tue.nl

Abstract. Business Process Management (BPM) systems provide a broad range of facilities to enact and manage operational business processes. Ideally, these systems should provide support for the complete BPM life-cycle: (re)design, configuration, execution, control, and diagnosis of processes. In the research presented, we evaluate the support provided by the FileNet P8 BPM Suite, which is consistently ranked as one of the leading commercial BPM systems. Taking realistic business scenarios as starting point, we completed a full pass through the BPM cycle with several tools from the FileNet P8 BPM Suite. We checked whether the expected support was provided by these tools and we also tested their interoperability. The outcome of our evaluation is that although strong support exists for the configuration, execution and control phase, process diagnosis and process redesign receive limited support. Interoperability exists between all phases, except between the diagnosis and the design phase.

1 Introduction

Business Process Management (BPM) systems can be seen as successors of Workflow Management (WFM) systems, which became popular in the mid-nineties. However, already in the seventies people were working on office automation systems which are comparable with today's WFM systems. Consider, for example, the OfficeTalk system developed by Ellis et al. at Xerox that was already able to support administrative processes based on Petri-net-based specifications of procedures [6]. Today, many WFM systems are available [2,11,13,14]. The core functionality of these systems can be described as *the ability to support an operational business process based on an explicit process model*, i.e., automating the "flow of work" without necessarily automating individual activities.

Recently, WFM vendors started to position their systems as BPM systems. We define BPM as follows: *Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information* [4]. This definition restricts BPM to operational processes, i.e., processes at the strategic level and processes that cannot be made explicit are excluded. It also follows that systems supporting BPM need to be "process aware". After all, without information about the operational processes at hand little support is possible. When comparing classical definitions of WFM [13] with the above definition of BPM, it can be observed that we assume BPM to offer a broader set of functionalities and support of the whole process life-cycle. This is also the "sales pitch" that many vendors use to market their products.

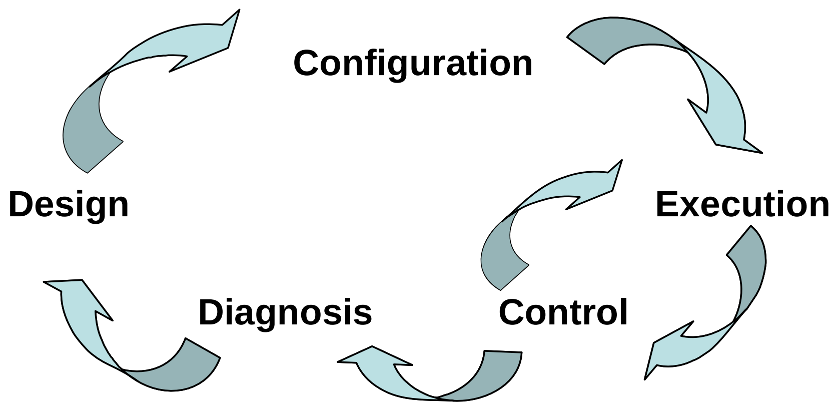

Fig. 1. The BPM life-cycle

The goal of this paper is to analyze whether today's BPM systems actually support the BPM life-cycle. To do this we use the BPM life-cycle as depicted in Figure 1. This life-cycle identifies five phases (*design*, *configuration*, *execution*, *control*, and *diagnosis*), which will be described later. The depicted life-cycle is a combination of the life-cycles presented in [4] and [20]. We will discuss the desired functionality in each of the phases. To make things more concrete, we have evaluated one particular system in detail: FileNet P8 BPM Suite (Version 3.5). We have selected this system because it is considered as one of the leading commercial BPM systems [7,8,9]. Moreover, the system is explicitly positioned by the vendor as a tool to support the whole BPM life-cycle.

We analyze the support of the FileNet P8 BPM Suite in each of the five phases shown in Figure 1. For our evaluation we performed a full pass through these phases using five realistic workflow scenarios, each including a concrete workflow process and life cycle context. We have used five workflows to be able to obtain additional insights when necessary. As starting point for our evaluation, we will assume that each workflow has already made one pass through the BPM cycle. The name and the related literature for each of the workflows is provided in Table 1. These particular workflows have been selected because the papers describing them provide a diagnosis of the improvement points and one or more alternative designs. Also, the original workflows and the alternatives have already been tested and the underlying data were available to us.

The remainder of this paper is organized as follows. First, we describe the BPM life-cycle in more detail and discuss the requirements that follow from it.

Table 1. The workflows used in our analysis

Workflow Name	Reference
Intake_Admin	Reijers, 2003 [18]
Credit application	Reijers, 2003 [18]
Intake_Meetings	Jansen-Vullers, Reijers, 2005 [12] ; Reijers, 2003 [18]
Bank account	Netjes, van der Aalst, Reijers, 2005 [15]
Mortgage request	van der Aalst, 2001 [1] ; Netjes, Vanderfeesten, Reijers, 2006 [17]

Then, in Section 3, we evaluate the FileNet P8 BPM Suite for each of the phases and in Section 4 we present our conclusions.

2 Evaluation approach based on the BPM life-cycle

In this section we discuss a system-independent approach to evaluate BPM systems. Pivotal to our evaluation approach is the BPM life-cycle depicted in Figure 1. Clearly, we want to evaluate the degree to which each phase is facilitated by a BPM system. Moreover, we want to asses the interoperability among phases, i.e., can information obtained or created in one phase be used in another phase? For example, a BPM system may incorporate a simulation tool, but it may be the case that the simulation model and the model used for execution are incompatible, forcing the user to re-create models or to set parameters twice.

First, we focus on the *design phase*. In case of an already existing process the goal of this phase is to create an alternative for the current process. This alternative should remedy the diagnosed weaknesses of the process according to the identified improvement possibilities. As indicated in Figure 1, this phase is in-between the diagnosis phase and the configuration phase, i.e., input from the diagnosis phase is used to identify improvement opportunities (e.g., bottlenecks or other weaknesses) and the output is transferred towards the configuration part of the BPM system. The resulting process definition consists of the following elements [3]:

- the process structure,
- the resource structure,
- the allocation logic, and
- the interfaces.

We would like to emphasize that a graphical editor by itself does not offer full support for the design phase. In the design phase the designer wants to experiment with designs, evaluate designs, and use input from the diagnosis phase. Some systems offer a simulation tool to support the design phase. Unfortunately, such a tool is often disconnected from the diagnosis phase, i.e., it is impossible to directly use historic data (e.g., to estimate service time distributions or routing probabilities). Moreover, simulation tools typically offer only what-if analysis, i.e., the designer has to come up with ideas for alternative designs and needs to analyze each alternative separately without sufficient tool support [17].

The *configuration phase* focuses on the detailed specification of the selected design. Note that in the design phase the emphasis is on the performance of the process, while in the configuration phase the emphasis shifts to the realization of the corresponding system. In principle, the design and configuration phase could use a common graphical editor, i.e., the configuration phase details the process definition created in the design phase. However, it is important (a) that the user is not forced to bypass the editor to code parts of the process and (b) that technical details do not need to be addressed in the design phase. If both phases use different tools or concepts, interoperability issues may frustrate a smooth transition from design to configuration.

In the *execution phase* the configured workflow becomes operational by transferring the process definition to the workflow engine. For the workflow execution not only the process definition data is required, but also context data about the environment with which the BPM system interacts. Relevant environmental aspects are:

- information on arriving cases,
- availability and behavior of internal/external resources and services.

The execution part of the BPM system captures the context data and relates it to specific instances of the workflow.

The execution of the operational business process is monitored in the *control phase*. The control part of the BPM system monitors on the one hand individual cases to be able to give feedback about their status and on the other hand, aggregates execution data to be able to obtain the current performance of the workflow. The monitoring of specific cases is done with the data from individual process executions without any form of aggregation, while obtaining the performance indicators requires aggregation of these data. Information about running cases can be used as input for the diagnosis phase. However, it can also be used to make changes in the process. For example, temporary bottlenecks do not require a redesign of the process, but require the addition of resources or other direct measures (e.g., not accepting new cases). Hence, the control phase also provides input for the execution phase.

In the *diagnosis phase* information collected in the control phase is used to reveal weaknesses in the process. In this phase the focus is usually on aggregated performance data and not on individual cases. This is the domain of process mining [5], business process intelligence [10], data warehousing, and classical data mining techniques. This diagnosis information is providing ideas for redesign (e.g., bottleneck identification) and input for the analysis of redesigns (e.g., historic data) in the design phase.

As indicated, it is not sufficient to support each of the five phases in isolation: interoperability among phases is vital for the usability of a BPM system. Consider for example the role of simulation. In a worst case scenario, a BPM system could offer a simulation tool that, on the one hand, cannot directly read the current workflow design used for execution (or relevant information is lost in some translation) and, on the other hand, cannot use any historic data to extract information about service times, routing probabilities, workloads, resource availability. Such a simulation tool probably offers little support for the BPM life-cycle [19].

3 Applying the Evaluation Approach to FileNet

We will evaluate the available BPM support by conducting a full pass through the BPM cycle with the aid of several tools from the FileNet P8 BPM Suite. We have evaluated the FileNet P8 BPM Suite, Version 3.5. The system has been used with Microsoft Windows 2000 as operating system, a Microsoft SQL Server as database, BEA Weblogic as J2EE application server and Microsoft Internet Explorer as browser. The P8 BPM Suite consists of six parts: Workflow Management, process design, process simulation, process tracking, process analysis and document review & approval (www.FileNet.com). The evaluation of FileNet's BPM abilities focuses on the tools supporting the first five parts. Document review & approval is not relevant for the evaluation; it only facilitates process management. In the remainder of this section, we consider FileNet's capabilities for each of the five BPM phases (design, configuration, execution, control, and diagnosis). A detailed illustration of the BPM support offered by FileNet can be found in [16] where we present the full pass through the BPM life-cycle for one of the five workflow scenarios.

3.1 Design

We start our evaluation with the design phase. For each of the five workflow scenarios mentioned in Table 1 we would like to create an alternative workflow with help from the FileNet P8 BPM Suite. We assume these workflows have already made one pass through the BPM cycle, meaning that the original workflow model and data from execution are present in the FileNet system. A workflow model for which an alternative should be made can be loaded in the FileNet *process designer*, which, however, does not support the creation of one or more alternatives. The redesign of the original model to obtain a better performing alternative should be done manually. For each of the workflows we take the alternatives described in the related paper and use the *process designer* to change the original model to the alternative model. One of the alternative designs made with the *process designer* is shown in Figure 2. The depicted design presents a medical process in which a mental patient is registered and assigned to medical employees (intakers), and for which intake meetings are planned. A detailed description of the process is available in [18]. More information on the modelling of workflows with the FileNet *process designer* can be found in [16].

The performance of each of the created alternatives should be evaluated to find the best alternative. For this we use the FileNet *process simulator*. For each alternative we create a simulation scenario for which we import the process

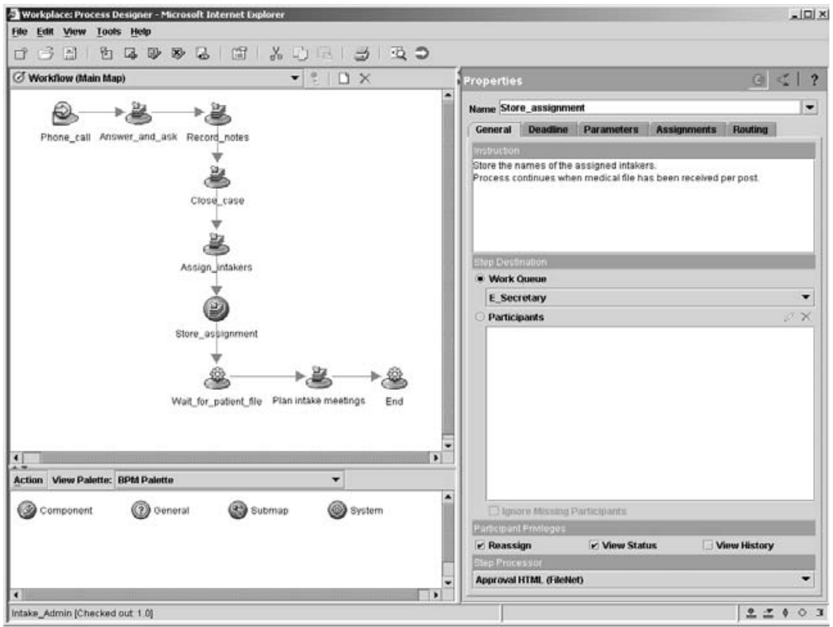

Fig. 2. Workflow model in the *process designer*

steps, their order and the allocation logic defined with the *process designer*. The imported data can not be changed in the *process simulator*, but a replacement can be imported from the *process designer* without the loss of settings. Other process definition data should be added to the simulation scenario manually. *Jobs* are connected to the process steps and assigned to *resources* which are allocated according to *shifts*. The notion of *shifts* allows for the scheduling of resources over the available working hours. Relating these *jobs*, *resources* and *shifts* to each other is rather complicated, because only one definition window can be open at the time and relations should also be indicated when defining a *job*, *resource* or *shift*.

In addition to the definition data there is context data required to perform a simulation. Historic data is present in the system, but it can only be used in a limited way. Historic information on arriving cases can be transferred to the *process simulator*, but all other data, like processing times and routing probabilities, should be derived from the execution data and included manually. It is only possible to provide constant values for the simulation parameters, so the simulation results will only provide a rough indication for the performance of a scenario. Simulation results are generated fast and with no additional effort. The use of the FileNet *process simulator* is in detail explained in [16]. A simulation scenario with simulation results is depicted in Figure 3. For each of the five workflows we choose the best alternative which we specify in detail in the configuration phase.

3.2 Configuration

The FileNet *process designer* is also used for the configuration of the chosen alternative workflows and offers interoperability between the design and the configuration phase. In the design phase we already specified the process structure and the mapping of resources to tasks for each workflow with the *process designer*. The more complicated parts of the process structure are detailed out in the configuration phase. Each workflow model contains one or more complex constructs, but besides one construct, we have been able to configure them all with the *process designer*. The resource structure, the allocation rules and the interfaces are defined outside the *process designer*. Defining outside the process designer allows for sharing with other processes, making the resource structure and the allocation rules reusable for other process definitions. All five workflows use the same allocation rules and some workflows have the same resource structure. The complete configuration of the five workflows, both inside and outside the *process designer* has been done in two working days. The configuration phase is strongly supported by the FileNet P8 BPM Suite.

As closure of the configuration phase, the workflow model is checked for completeness by the system and a workflow instance could be launched to pretest the execution of the workflow. Another possible check would have been a check on the correctness of the model, conform the verification of workflow processes provided by the Woflan tool [21], but such a verification is not supported by the FileNet system. The configuration of the workflows is necessary for their execution.

3.3 Execution

The execution phase is started with the transfer of the workflow configurations to the FileNet *process engine*. All process definition data are transferred to the *process engine* providing interoperability between the configuration and the execution phase. Resources work on the processes in operation via an inbox. The FileNet P8 BPM Suite offers integration with external applications, document management, integration with content management, and interaction between inter-related processes. The FileNet system supports the execution phase in an excellent way. We expected mature support for execution, because this support has traditionally been the heart of a WFM system and many systems provide extended support for the execution phase. In the execution phase context data is related to each specific instance of a workflow and this combination of definition and context data is used for the control of a workflow.

3.4 Control

In the control phase, the operational business process is monitored to follow individual cases and to obtain the performance of a workflow. The first way of monitoring is supported by the FileNet *process administrator* and the second by the *analysis engine*, providing a strong support for the control phase.

The execution data for individual cases and other workflow events are logged by the *process engine*. The history of a certain workflow, step or work item can be tracked in the log through the FileNet *process administrator*. For the workflows with conditional routing this gives the opportunity to determine which steps were executed for a specific case. With the *process administrator* it can also be determined how certain decisions were made during execution allowing us to see at which point and why a certain case was rejected.

The performance of a workflow is read from aggregated execution data. The execution data present in the *process engine* is aggregated and parsed to the FileNet *analysis engine*. Interoperability exists between the execution and the control phase, because all execution data necessary for control are available either through the *process engine* or the *analysis engine*. The aggregated performance data resides on a separate engine to not affect the performance of the *process engine*. Reporting and analysis of the aggregated data is facilitated by twenty out-of-the-box reports; each graphically presenting the data related to one performance indicator. It is possible to specify custom reports, but this requires advanced Excel skills. The representation of the data can be manipulated by adjusting the detail level or by filtering the data.

An analysis of the work present in the queues gives insight in the existence of temporary bottlenecks in the process. This information is used as feedback for the execution phase. The feedback, however, is obtained from human interpretation of the analysis results and does not contain suggestions for the removal of the bottleneck. More permanent weaknesses in the process could also be revealed based on the analysis of performance data and this is done in the diagnosis phase.

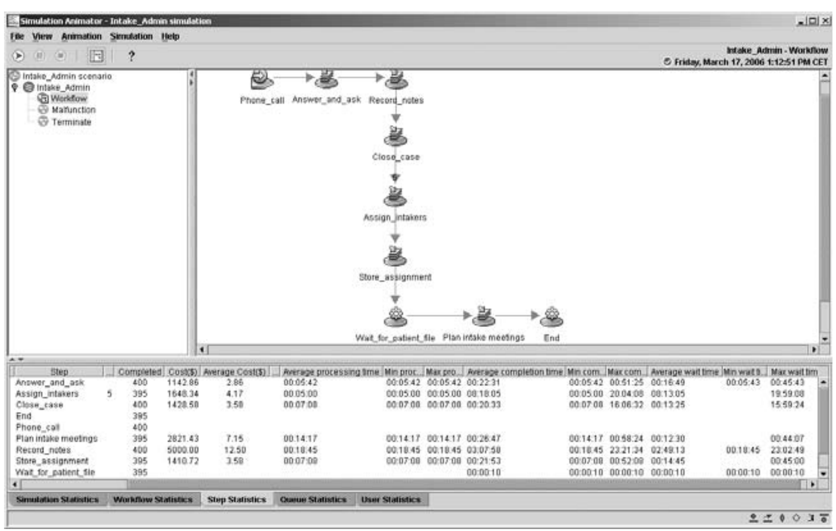

Fig. 3. Simulation results from the *process simulator*

3.5 Diagnosis

In the diagnosis phase, problems and improvement possibilities are identified through analysis of the operational process. The *analysis engine* facilitates the control and the diagnosis phase, creating interoperability between the two phases. Analysis reports present an aggregated view on the performance data and weaknesses in the process are derived from this. The derivation, however, is not supported by the FileNet P8 BPM Suite and is based on human insights. A system not capable of identifying process weaknesses is certainly unable to provide improvement suggestions for these weaknesses. The FileNet P8 BPM Suite provides limited support for the diagnosis phase and the creation of ideas for process improvement should be done manually.

The ideas for redesign generated in the diagnosis phase could result in another pass through the BPM cycle starting with a new design phase. When we started our pass in the design phase it became clear that historic performance data is necessary to obtain the performance of the created redesigns with simulation. We already mentioned that only historic arrival data could be used, making the interoperability between the diagnosis and the design phase limited. We did not mention yet that data generated with simulation can also be transferred to the *analysis engine* and presented in the performance reports. This provides a comprehensive view on the simulation results. Nevertheless, presenting the correct data becomes problematic when multiple scenarios of the same simulation model have been simulated over the same simulation time. It is not possible to select the data of only one of the scenarios, while the aggregation of all simulation data leads to unusable results. The only solution for this is clearing the *analysis engine* before each new simulation run, which does not only lead to unworkable situations, but will also remove the historic execution data from the *analysis engine*.

4 Conclusions

The conclusions from this study are summarized in Table 2. In Table 2 we present the support required for each phase in the BPM life-cycle and the support provided by the FileNet P8 BPM Suite. From our evaluation we conclude that FileNet provides strong support for the configuration, the execution and the control phase. In particular,

- The configuration phase is well supported by the *process designer*.
- The execution of the workflow is strongly supported by the *process engine*.
- The control phase is supported by the *process administrator* and the *analysis engine*.

Less explicit support is available for the diagnosis and design phase. Some support in the diagnosis phase is provided by the *process analyzer*, which gives an aggregate view on the data. However, the search for weaknesses in the process is not supported and certainly no improvement suggestions are generated. Furthermore, in the design phase the creation of the alternatives is not supported.

Limited support is available through the representation of the alternatives as facilitated by the *process designer* and the selection of the best alternative by the *process simulator*.

Table 2. Summary of the evaluation

Phase	Required support	FileNet support
Design	Make redesign	-
	Model designs	Process designer
	Evaluate designs	Process simulator
	Compare designs	-
	Input from diagnosis phase available	- (only arrival data)
	Output for configuration phase available	Through process designer
Configuration	Model detailed designs	Process designer
	Input from design phase available	Through process designer
	Output for execution phase available	Transfer of process definition
Execution	Workflow engine	Process engine
	Capture context data	Process engine
	Input from configuration phase available	Transfer to process engine
	Output for control phase available	Transfer from process engine
Control	Monitor specific cases	Process administrator
	Aggregation of execution data	Analysis engine
	Monitor performance	Process analyzer
	Input from execution phase available	Transfer to analysis engine
	Output for diagnosis phase available	Through analysis engine
	Output for execution phase available	-
Diagnosis	Reveal weaknesses	Process analyzer
	Identify improvement points	-
	Input from control phase available	Through analysis engine
	Output for design phase available	- (only arrival data)

- : not supported by FileNet, should be done manually.

The conclusion for our interoperability evaluation is that the interoperability of the FileNet process tools is notably supported in the transitions between the design, the configuration, the execution, the control and the diagnosis phase. At the same time, the interoperability between the diagnosis and the design phase is limited to the use of historic arrival data (present in the *analysis engine*) for the simulation. All other performance data present in the *analysis engine* can not be passed to the *process simulator* and should be copied manually. Although interoperability exists between the execution and control phase, the loop back from control to execution is not supported. In the control phase temporary bottlenecks can be identified, but human intervention is required to interpret the findings and tune the operational process.

These insights are in line with the support that could be expected from a WFM system, as these systems are well-known for their emphasis on the configuration, execution and control phase. Nonetheless, it is also clear that opportunities exist to improve the support that so-called BPM systems offer to execute the entire BPM life-cycle. We consider the FileNet P8 BPM suite as a relevant benchmark for many of the other available systems, because of its broad range of features and market dominance. The improvement opportunities also set the stage for further research, which in our view should focus on transforming available BPM theory into BPM system support. In particular, our future work will focus on addressing the gap between redesign theory and practice with the development of redesign tools [17].

Acknowledgement

We would like to thank the consultancy and IT support staff from FileNet for their kind assistance in carrying out this study. This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs.

References

1. W.M.P. van der Aalst. Reengineering Knock-out Processes. *Decision Support Systems*, 30(4):451–468, 2001.
2. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
3. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods and Systems (in Chinese)*. Tsingua University Press, Beijing, China, 2004.
4. W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske. Business Process Management: A Survey. In W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, editors, *International Conference on Business Process Management (BPM 2003)*, volume 2678 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, Berlin, 2003.
5. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.
6. C.A. Ellis. Information Control Nets: A Mathematical Model of Office Information Flow. In *Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems*, pages 225–240, Boulder, Colorado, 1979. ACM Press.
7. Gartner. Gartner's Magic Quadrant for Pure-Play BPM. http://www.gartner.com, 2003.
8. Gartner. Gartner's Magic Quadrant for Pure-Play BPM. http://www.gartner.com, 2004.
9. Gartner. Gartner's Magic Quadrant for Pure-Play BPM. http://www.gartner.com, 2005.
10. D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.C. Shan. Business process intelligence. *Computers in Industry*, 53(3):321–343, 2004.
11. S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation*. International Thomson Computer Press, London, UK, 1996.

12. M.H. Jansen-Vullers and H.A. Reijers. Business Process Redesign at a Mental Healthcare Institute: A Coloured Petri Net Approach. In K. Jensen, editor, *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005)*, volume 576 of *DAIMI*, pages 21–38, Aarhus, Denmark, October 2005. University of Aarhus.
13. P. Lawrence, editor. *Workflow Handbook 1997, Workflow Management Coalition*. John Wiley and Sons, New York, 1997.
14. M. zur Muehlen. *Workflow-based Process Controlling: Foundation, Design and Application of workflow-driven Process Information Systems*. Logos, Berlin, 2004.
15. M. Netjes, W.M.P. van der Aalst, and H.A. Reijers. Analysis of Resource-Constrained Processes with Colored Petri Nets. In K. Jensen, editor, *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005)*, volume 576 of *DAIMI*, pages 251–266, Aarhus, Denmark, October 2005. University of Aarhus.
16. M. Netjes, H.A. Reijers, and W.M.P. van der Aalst. FileNet's BPM life-cycle support. BPM Center Report BPM-06-07, BPMcenter.org, 2006.
17. M. Netjes, I. Vanderfeesten, and H.A. Reijers. "Intelligent" Tools for Workflow Process Redesign: A Research Agenda. In C. Bussler and A. Haller, editors, *Business Process Management Workshops: BPM 2005*, volume 3812 of *Lecture Notes in Computer Science*, pages 444–453. Springer-Verlag, Berlin, 2006.
18. H. Reijers. *Design and Control of Workflow Processes: Business Process Management for the Service Industry*, volume 2617 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2003.
19. H.A. Reijers and W.M.P. van der Aalst. Short-Term Simulation: Bridging the Gap between Operational Control and Strategic Decision Making. In M.H. Hamza, editor, *Proceedings of the IASTED International Conference on Modelling and Simulation*, pages 417–421. IASTED/Acta Press, Anaheim, USA, 1999.
20. H.A. Reijers and K.M. van Hee. Business Process Management in een Notendop (in Dutch). *VIP: Vakblad voor Documentmanagement*, 10(6):30–32, 2004.
21. H.M.W. Verbeek, T. Basten, and W.M.P. van der Aalst. Diagnosing Workflow Processes using Woflan. *The Computer Journal*, 44(4):246–279, 2001.