

# Questioning BPM?

Respondent name: Wil van der Aalst

## Question #4

To understand the current business process market, it is good to consider the roots of today's *Business Process Management (BPM) systems*. In the seventies, people like Skip Ellis, Anatol Holt, and Michael Zisman already worked on so-called *Office Information (OI) systems*, which were driven by explicit process models. Ellis et al. developed OI prototype systems such as Officetalk-Zero and Officetalk-D at Xerox PARC in the late 1970-ties. These systems used variants of Petri nets to model processes. Another example from the same period is SCOOP (System for Computerizing of Office Processes), developed by Michael Zisman. SCOOP also used Petri nets to represent business processes. Officetalk, SCOOP and other OI systems were created in a time where workers were typically not connected to a network. Consequently, these systems were not widely adopted. Nevertheless, it is good to realize that the vision still driving today's BPM systems was already present in the late 1970-ties.

A second wave of BPM-like systems emerged in the mid-nineties. Numerous vendors started to offer generic *Workflow Management (WFM) systems*. There was the expectation that WFM systems would get a role comparable to Database Management (DBM) systems, i.e., data would be subcontracted to DBM systems and processes would be subcontracted to WFM systems. DBM systems managed to become an integral part of almost all information systems since the 1970-ties. However, WFM systems were not widely adopted and (unlike DBM systems) did not manage to become an integral part of the typical information system of an organization. The early WFM systems were focusing too much on automation, not acknowledging the management aspects and the need for flexibility. Processes can also be captured using conventional programming languages. Indeed, most workflows are hard-coded and hidden in application programs.

As mentioned, one of the pitfalls of the classical WFM systems was the *lack of management support*. The focus was on automa-

tion rather than the ability to analyze, manage, and control systems. At the turn of the century, a third wave of BPM-like systems emerged. These systems had a broader scope than WFM-technology: From process automation and process analysis to operations management and the organization of work. The trend to provide better management support is still ongoing in current systems. Also there have been various attempts to make these systems more flexible than WFM systems. Declarative BPM/WFM systems and case management approaches signify this trend towards more flexibility.

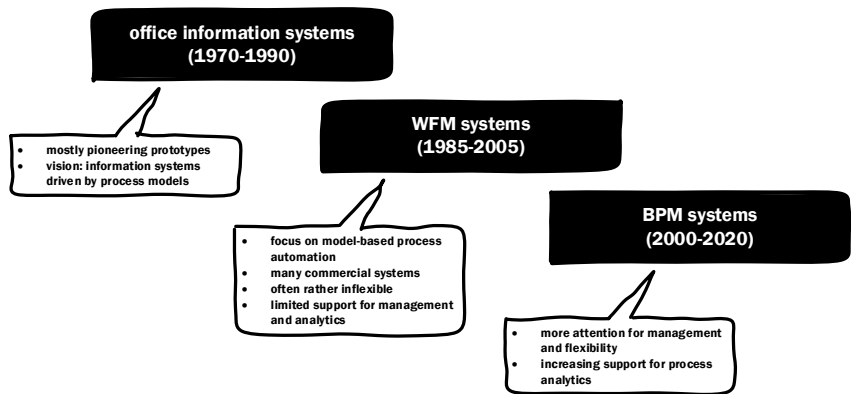


Figure 1.1: Three "waves" of process-aware information systems

BPM should aim at improving operational business processes, *with or without* BPM systems. For example, by modeling a business process and analyzing it using simulation, management may get ideas on how to reduce costs while improving service levels. It is often not necessary to introduce a full-fledged BPM system. More important, BPM approaches should exploit the event data widely available in today's organizations. Many BPM authors (including Smith and Fingar in their 2003 book) fail(ed) to see the im-

portance of business process intelligence and process mining. Discussions on the definition of BPM and the references to the pi calculus now seem silly. Most process improvements and innovations are driven by data. Unfortunately, processes are not at the forefront in most data science and Big data initiatives. The BPM community should take on the challenge to make these initiatives more process-centric.

## Question #12

BPM requires close collaboration between IT specialists, management, domain experts, and workers. Processes need to be modeled, enacted, analyzed, and managed. Figure 1.2 shows the four main types of activity in any larger BPM effort: *model* (creating a process model to be used for analysis or enactment), *enact* (using a process model to control and support concrete cases), *analyze* (analyzing a process using a process model and/or event logs), and *manage* (setting goals, adjusting the process, reallocating resources, controlling, etc.). Each of these activities combines IT aspects with business/organizational aspects. One cannot expect all involved actors to be BPM experts. However, one would expect that self-proclaimed BPM professionals to master all the basics of process management. For example, it is remarkable that basic concepts such as the workflow patterns and the difference between process/activity instance and type are not well understood by many BPM professionals. Solid BPM education is of the utmost importance to improve maturity of our discipline.

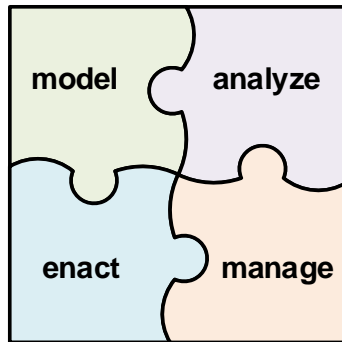


Figure 1.2: BPM professionals should be educated to fully understand the IT aspects with business/organizational aspects of the four main puzzle pieces of BPM.

In the context of BPM we can identify at least three groups of actors:

1. **BPM specialists:** The group of people full-time involved in BPM and part of some central group (for example a Center of Excellence) not connected to a specific process and not involved in the actual operational processes. Ideally, BPM specialists have an academic degree in computer science or industrial engineering with a special focus on process management and business information systems.
2. **BPM facilitators:** The group of people involved in BPM projects, but not considered to be BPM specialists (working only part-time on BPM). People in this group need to be able to model processes, define KPIs, use BI tools, and understand the IT-implications of process change. BPM facilitators are added to project teams that aim to improve existing processes or develop new processes. Certification or internal training programs can be used to ensure sufficient BPM knowledge.
3. **BPM-aware managers and workers:** The group of people that use the fruits of BPM. These are workers

performing the actual activities in the processes and the people managing them. People in this group just need to be aware of BPM concepts.

In terms of required skills, major changes can be anticipated due to the increased availability of process-related event data. Developments related to Big data (in the broadest sense), will impact process management. BPM professionals will need to acquire more data-science skills. BPM will become more "evidence based" and less subjective.

In recent years, "data science" has become a common term to refer to the emerging discipline revolving around the abundance of data in modern organizations. We would like to confront "data science" with the umbrella term "process science" which refers to the broader discipline that combines knowledge from information technology and knowledge from management science to improve and run operational processes.

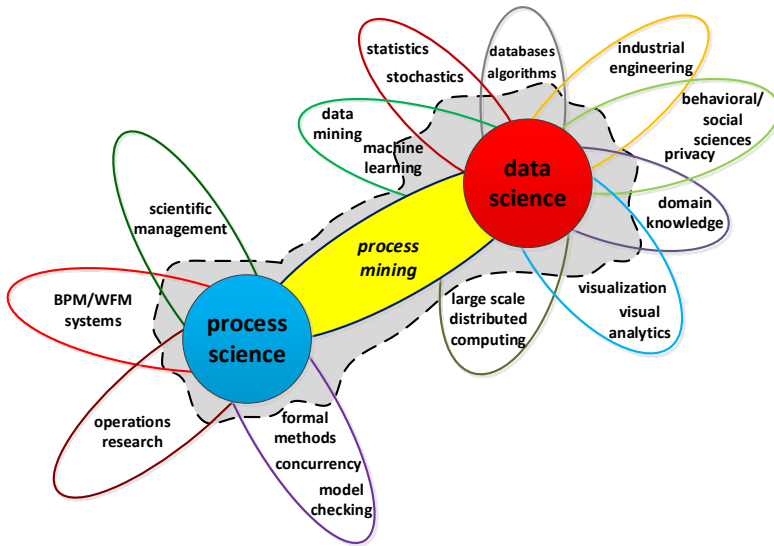


Figure 1.3: BPM professionals should combine "process science" and "data science" skills

Figure 1.3 shows some of the key ingredients of both data science and process science. BPM professionals should master the basics of both groups of ingredients. For example, basic knowledge of large-scale distributed computing is needed when participating in Big data projects. Knowledge of visualization techniques is beneficial when presenting the effects of process improvement projects. Figure 1.3 also shows that process mining is the bridge between process science and data science. Data science approaches tend to be process agnostic whereas process science approaches tend to be model-driven without considering the "evidence" hidden in the data. Obviously, both worlds need to be connected and integrated and process mining is one of the key technologies to achieve this.

*Process mining* aims to discover, monitor and improve real processes by extracting knowledge from event logs readily available in today's information systems. Starting point for process mining is



an *event log*. Event data can be used to conduct three types of process mining: *process discovery* (finding out what is really happening in the process and representing this as a process model ready for analysis), *conformance checking* (understanding where and why processes deviate and whether these deviations are harmful), and *enhancement* (extending models with performance and conformance information and generating process improvement ideas).

Interestingly, process mining (and other data-driven BPM technologies) will help to improve collaboration between IT specialists, management, domain experts, and workers. The moment the *real* processes are properly visualized, discussion become more focused and fact-driven.

## Question #14

Process modeling *can* be a valuable activity. Often the activity of modeling is more important than the resulting models. Modeling forces people to elicit assumptions, requirements, and goals. Moreover, process modeling is a great tool to stimulate discussions on process improvements. However, too much energy is wasted on notational issues. BPM professionals tend to be almost religious about the process notations they use. BPMN, Petri nets, Workflow nets, EPCs, BPEL, and UML activity diagrams are just a few examples of the many languages used. Debates on which notation to use are seldom productive and definitely do not contribute to concrete process improvements. In recent years, BPMN has become the de-facto standard for process modeling, but people tend to use only a tiny subset of the BPMN language (the lion's share of BPMN modeling elements are unknown or deemed irrelevant). Moreover, many BPM professionals lack good modeling skills. This is reflected by the many errors in process models. Models have internal inconsistencies such as deadlocks and live-locks or different types of process instances are mixed in the same diagram (e.g., activities related to customer orders and order-line items are connected thus blurring the instance notion).

The return on investment of modeling depends on (1) the quality of the team making the models and (2) the BPM maturity or ambition level of the organization. Flawed models have little added value, and will not be taken seriously. Moreover, organizations at one of the lower BPM maturity levels tend to use process models merely as "wallpaper". If processes are documented without a clear purpose and the models tend to be outdated, then modeling is a waste of time. Models are most effective when they are used *actively*, for example to drive a BPM system or to project event data on. Models should also be used *continuously*. Models can be compared to (geographic) maps. There is no need to make

maps without actively using them in day-to-day operations. The constant confrontation of maps with reality will help to keep them up-to-date. If organizations are not taking models seriously, the "return on modeling" will be limited.

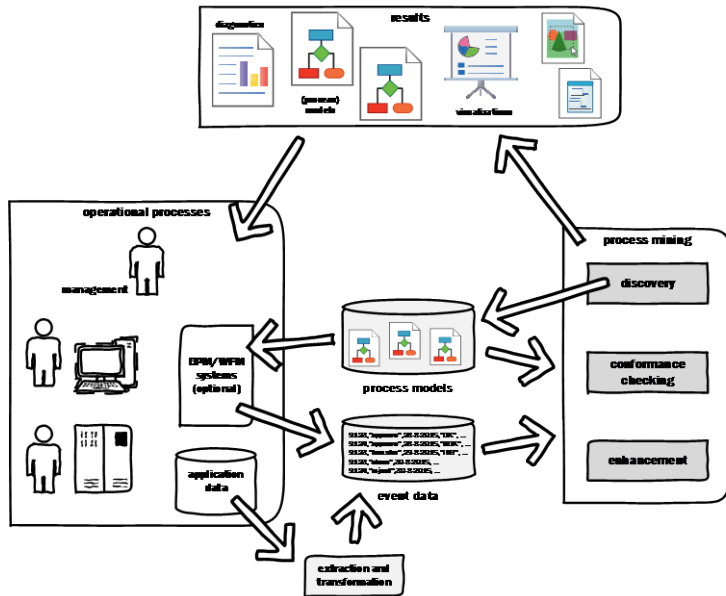


Figure 1.4: Positioning process mining as an alternative to pure modeling approaches.

Compared to business process modeling, data-driven BPM approaches such as process mining are likely to provide a higher return on investment. In a time where organizations are drowning in event data, it seems foolish to model processes by hand not considering the "evidence" in databases, transaction logs, and audit trails. The growing importance of Big Data and Data Science signify a trend towards more data-driven approaches. Data should not just be used *within* processes, but also to learn more *about* processes. Recent breakthroughs in *process mining* research make it possible to discover, analyze, and improve business processes based on event data. Process mining results can be viewed as X-

rays showing what is *really* going on *inside* processes. Such X-rays can be used to diagnose problems and suggest proper treatment.

Process mining is likely to provide higher returns on investment than traditional process modeling approaches: the threshold to get started is low and process mining results are *real*. Unlike modeling, process mining provides insights and diagnostics based on *facts*. Through process mining, process models can be "confronted" with the actual processes (through event data). As a result, process models do not end up in drawers or serve as wallpaper. Visualizing detailed performance and conformance diagnostics on process models on a day-to-day basis is a potential key asset for any BPM initiative. It is a great tool to make workers aware of BPM. Unfortunately, traditional BPM professionals (the "BPM dinosaurs") have difficulties transitioning to more data-centric forms of BPM.

## Question #4

Effective business processes must be able to accommodate changes in the environment in which they operate, e.g., new laws, changing workloads, changes in business strategy, or emerging technologies. The ability to support such changes is definitely a key concern of BPM. Unsurprisingly, flexibility has been one of the hot topics in BPM/WFM research since the mid 1990-ties. Recent discussions on case management, iBPM, CMMN, etc. tend to forget about seminal work done in the last two decades. Many approaches have been proposed, some more mature than others. In general there is a huge gap between the claims of WFM/BPM vendors and actual flexibility support provided by their systems. Newcomers to the field should have an open eye for novel flexibility approaches, but also understand why previous approaches failed. It is also crucial to realize that solutions cannot be found in new standardization proposals like CMMN. Moreover, discussions on notation (BPMN versus XYZ) will not assist in capturing the complexity and dynamicity of real processes!

To put things in a historic perspective and to provide pointers for newcomers in the field, let us consider a few papers related to flexibility in BPM:

- [1] C.A. Ellis, K. Keddara, and G. Rozenberg. *Dynamic Change within Workflow Systems*. Proceedings of the Conference on Organizational Computing Systems, pages 10-21, Milpitas, California, August 1995. ACM SIGOIS, ACM Press, New York.
- [2] M. Reichert and P. Dadam. *ADEPTflex: Supporting Dynamic Changes of Workflow without Loosing Control*. Journal of Intelligent Information Systems, 10(2):93-129, 1998.
- [3] W.M.P. van der Aalst and P.J.S. Berens. *Beyond Workflow Management: Product-Driven Case Handling*. International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2001), pages 42-51. ACM Press, New York, 2001.

- [4] W.M.P. van der Aalst, M. Weske, and D. Grünbauer. *Case Handling: A New Paradigm for Business Process Support*. Data and Knowledge Engineering, 53(2):129-162, 2005.
- [5] M. Pesic, M. H. Schonenberg, N. Sidorova, and W.M.P. van der Aalst. *Constraint-Based Workflow Models: Change Made Easy*. Proceedings of the OTM Conference on Cooperative information Systems (CoopIS 2007), volume 4803 of Lecture Notes in Computer Science, pages 77-94. Springer-Verlag, Berlin, 2007.
- [6] W.M.P. van der Aalst, M. Pesic, and H. Schonenberg. *Declarative Workflows: Balancing Between Flexibility and Support*. Computer Science - Research and Development, 23(2):99-113, 2009.
- [7] W.M.P. van der Aalst and S. Jablonski. *Dealing with Workflow Change: Identification of Issues and Solutions*. International Journal of Computer Systems, Science, and Engineering, 15(5):267-276, 2000.
- [8] W.M.P. van der Aalst. *Business Process Management: A Comprehensive Survey*. ISRN Software Engineering, pages 1-37, 2013.

In [1] the problem of workflow flexibility was first discussed at a level allowing to reason about the foundational limits of change. The "dynamic change bug" was identified. Since 1995 many WFM systems have been developed to provide flexibility and to deal with phenomena like the "dynamic change bug". The ADEPT system [2] developed at the University of Ulm is probably the system that provided and still provides the most powerful flexibility features. Systems like ADEPT are still driven by procedural models, i.e., process notations similar to BPMN, Petri nets, Workflow nets, EPCs, and UML Activity Diagrams. Alternative (non-procedural) approaches that are interesting to consider include: (1) data-driven *case handling approaches* [3,4] and (2) *declarative approaches* [5,6]

*Case handling* approaches [3,4] supported by systems such as BPM|one (Perceptive) and the earlier FLOWer (Pallas Athena) emerged around the turn of the century. The core features of case

handling, as defined in [3,4], can be described as follows:

- *avoid context tunneling* by providing all information available (i.e., present the case as a whole rather than showing just bits and pieces),
- decide which activities are enabled on the basis of the information available rather than the activities already executed (i.e., processes are *data-driven* rather than control-flow centric),
- *separate work distribution from authorization* and allow for additional types of roles, not just the execute role (e.g., skip, redo, etc.),
- allow workers to view and add/modify data *before* or *after* the corresponding activities have been executed (e.g., information can be provided the moment it becomes available).

These case handling features aim to provide the flexibility needed. This way the pitfalls of traditional production-style WFM systems can be avoided. The lengthy debates on case management conducted between 2000 and 2005 (roughly one decade (!) after the definition of case handling [3,4]) did not clarify the understanding of case management/handling. Numerous parties provided alternative definitions for the term case management. However, the impact on BPM products was limited and the "new" functionalities provided were often trivial and far from surprising (comparable to the "goto" in programming).

*Declarative approaches* such as *Declare* [5,6] aim for a better balance between support and flexibility. Traditional approaches use procedural process models to specify the execution procedure explicitly (i.e., step-by-step). *Declare* is based on *constraints*, i.e., *anything is possible as long as it is not explicitly forbidden*. Constraint-based models *implicitly* specify the execution procedure by listing a collection of (hard or soft) constraints: Any process execution that does not violate the constraints is possible. Languages like *Declare* are typically grounded in some temporal logic (e.g., LTL) to formulate

behavioral constraints. It is possible to specify that an activity should always be preceded by another one or that two activities should never be executed for the same case. Standardization proposals like OMG's Case Management Model and Notation (CMMN) claim to be declarative, but it is less clear what this means. In fact there is controversy about the level of difference between BPMN and CMMN, and whether the standards should be merged or not.

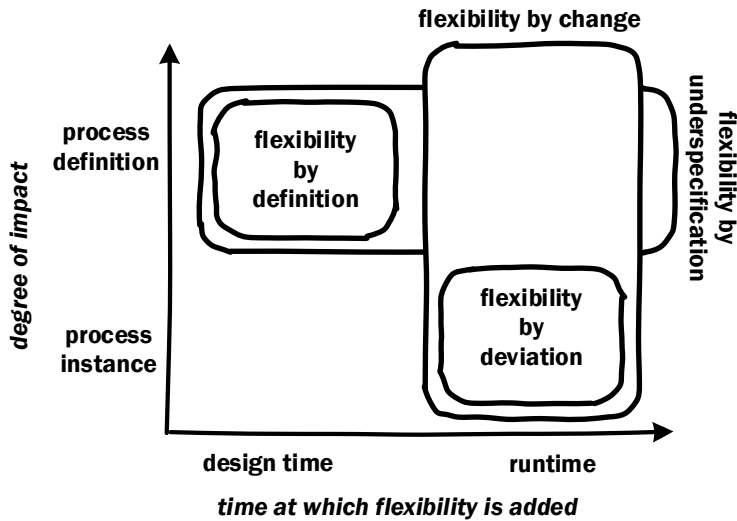


Figure 1.5: Taxonomy of process flexibility identifying four main flexibility types: (1) flexibility by definition, (2) flexibility by deviation, (3) flexibility by underspecification, and (4) flexibility by change.

Rather than debating on standards, it seems more worthwhile to provide a taxonomy of process flexibility based on [7,8] here. The taxonomy shown in Figure 1.5 identifies four main flexibility types: (1) *flexibility by definition*, (2) *flexibility by deviation*, (3) *flexibility by underspecification*, and (4) *flexibility by change*.

*Flexibility by definition* is the ability to incorporate alternative



execution paths within a process definition at design time such that selection of the most appropriate execution path can be made at runtime for each process instance. For example, parallelism defined at design time leaves the actual ordering of activities open and thus provides more flexibility than sequential routing. All WFM/BPM systems support this type of flexibility. However, declarative languages make it easier to defer choices to runtime (anything is possible unless there is a constraint preventing it).

*Flexibility by deviation* is the ability for a process instance to deviate at runtime from the execution path prescribed by the original process without altering the process definition itself. The deviation can only encompass changes to the execution sequence for a specific process instance, and does not require modifications of the process definition. The case handling approach [3,4] supported by systems like BPM|one allows for such forms of flexibility. It is possible to undo, redo, and skip an activity. Moreover, data can be entered earlier or later because the state is continuously recomputed based on the available data.

*Flexibility by underspecification* is the ability to execute an incomplete process specification, i.e., a model that does not contain sufficient information to allow it to be executed to completion. An incomplete process specification contains one or more so-called placeholders. These placeholders are nodes which are marked as underspecified (i.e., "holes" in the specification) and whose content is specified during the execution of the process. The underspecified parts are filled in through late binding or late modeling.

*Flexibility by change* is the ability to modify a process definition at run-time such that one or all of the currently executing process instances are migrated to a new process definition. Changes may be introduced both at the process *instance* and the process *type* levels. A *momentary change* (also known as change at the instance level) is a change affecting the execution of one or more selected process instances. An *evolutionary change* (also known as change at the type level) is a change caused by modification of the process definition, potentially affecting all new process instances. A typical

example of the evolutionary change is the redesign of a business process to improve the overall performance characteristics by allowing for more concurrency. Running process instances that are impacted by an evolutionary or a momentary change need to be handled properly. If a running process instance is transferred to the new process, then there may not be a corresponding state (called the "dynamic change bug" mentioned earlier). See ADEPT [2] and Declare [5,6] for approaches supporting a wide range of process changes.

For each of the four types of flexibility identified in Figure 1.5, there exists a range of articles and (prototype) systems. These show that dynamic and complex processes are difficult to support. Instead of proposing new standards like CMMN, the BPM community should offer systems that actually support flexibility and provide empirical evidence for it.

## Short bio

Prof.dr.ir. Wil van der Aalst is a full professor of Information Systems at the Technische Universiteit Eindhoven (TU/e). At TU/e he is the scientific director of the Data Science Center Eindhoven (DSC/e). Since 2003 he holds a part-time position at Queensland University of Technology (QUT). His personal research interests include workflow management, process mining, Petri nets, business process management, process modeling, and process analysis. Wil van der Aalst has published more than 180 journal papers, 18 books (as author or editor), 400 refereed conference/workshop publications, and 60 book chapters. Many of his papers are highly cited (he one of the most cited computer scientists in the world and has an H-index of 119 according to Google Scholar) and his ideas have influenced researchers, software developers, and standardization committees working on process support. He has been a co-chair of many conferences including the Business Process Management conference, the International Conference on Cooperative Information Systems, the International conference on the Application and Theory of Petri Nets, and the IEEE International Conference on Services Computing. He is also editor/member of the editorial board of several journals, including Computing, Distributed and Parallel Databases, Software and Systems Modeling, the International Journal of Business Process Integration and Management, the International Journal on Enterprise Modelling and Information Systems Architectures, Computers in Industry, Business & Information Systems Engineering, IEEE Transactions on Services Computing, Lecture Notes in Business Information Processing, and Transactions on Petri Nets and Other Models of Concurrency. In 2012, he received the degree of doctor honoris causa from Hasselt University in Belgium. He served as scientific director of the International Laboratory of Process-Aware Information Systems of the National Research University, Higher School of Economics in Moscow.

In 2013, he was appointed as Distinguished University Professor of TU/e and was awarded an honorary guest professorship at Tsinghua University. In 2015, he was appointed as honorary professor at the National Research University, Higher School of Economics in Moscow. He is also a member of the Royal Netherlands Academy of Arts and Sciences (Koninklijke Nederlandse Akademie van Wetenschappen), Royal Holland Society of Sciences and Humanities (Koninklijke Hollandsche Maatschappij der Wetenschappen) and the Academy of Europe (Academia Europaea).