Process-Aware Information Systems: Lessons to Be Learned from Process Mining

Wil M.P. van der Aalst

Department of Mathematics and Computer Science,
Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
w.m.p.v.d.aalst@tue.nl

Abstract. A *Process-Aware Information System* (PAIS) is a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models. Example PAISs are workflow management systems, case-handling systems, enterprise information systems, etc. This paper provides a brief introduction to these systems and discusses the role of process models in the PAIS life-cycle. Moreover, it provides a *critical reflection on the state-of-the-art based on experiences with process mining*. Process mining techniques attempt to extract non-trivial and useful information from event logs. One aspect of process mining is control-flow discovery, i.e., automatically constructing a process model (e.g., a Petri net) describing the causal dependencies between activities. The insights provided by process mining are very valuable for the development of the next generation PAISs because they clearly show a mismatch between the models proposed for driving these systems and reality. On the one hand, models tend to oversimplify things resulting in systems that are too restrictive. On the other hand, models fail to capture important aspects of business processes.

1 Introduction

In the last two decades there has been a shift from "data-aware" information systems to "process-aware" information systems [24]. To support business processes, an enterprise information system needs to be aware of these processes and their organizational context. Early examples of such systems were called WorkFlow Management (WFM) systems [4,28,33,36,38,41,45,67]. In more recent years, vendors prefer the term Business Process Management (BPM) systems. BPM systems have a wider scope than the classical WFM systems and are not just focusing on process automation. BPM systems tend to provide more support for various forms of analysis (e.g., simulation) and management support (e.g., monitoring). Both WFM and BPM aim to support operational processes that are often referred to as "workflow processes" or simply "workflows". We will use the generic term *Process-Aware Information System* (PAIS) to refer to systems that manage and execute such workflows.

K. Jensen and W. van der Aalst (Eds.): ToPNoC II, LNCS 5460, pp. 1–26, 2009.

In a Service Oriented Architecture (SOA) the information system is seen as a set of connected services. A PAIS can be realized using such an architecture and in fact it is very natural to see processes as the "glue" connecting services. The fit between SOA and PAIS is illustrated by emerging standards such as BPEL [20] and BPMN [68]. The focus on web services and SOA has stirred up enthusiasm for process-orientation. As a result it is expected that in the future generic PAISs will start to play a more important role. However, at the same time it should not be forgotten that most PAISs are dedicated towards a particular application domain or even a specific company.

The flow-oriented nature of workflow processes makes the Petri net formalism a natural candidate for the modeling and analysis of workflows. Most workflow management systems provide a graphical language which is close to Petri nets. Although the routing elements are different from Petri nets, the informal semantics of the languages used are typically token-based and hence a (partial) mapping to Petri nets is relatively straightforward. This explains the interest in applying Petri nets to PAISs.

The purpose of this paper is twofold. On the one hand, we aim to provide an *introduction to PAISs and the role of models in the development and configuration* of such systems. On the other hand, we would like to share some *insights obtained through process mining*. Process mining exploits event logs of real processes and uses these to discover models or check the conformance of existing ones. Experiences with process mining show that there are typically large discrepancies between the idealized models used to configure systems and the real-life processes. Moreover, process mining has changed our perception of models. For example, there is no such thing as *the* model. In any situation different models are possible all providing a particular view on the process at hand. Based on our experiences using process mining, we would like to challenge some of the basic assumptions related to PAIS and business process modeling.

The remainder of this paper is organized as follows. Section 2 provides a definition and classification of PAISs. The role of process models is discussed in Section 3 and Section 4 briefly introduces the concept of process mining. Section 5 presents the lessons that can be learned from process mining. This section serves as a "reality check" for PAIS research. Section 6 concludes the paper.

2 Process-Aware Information Systems

In this paper we adopt the following definition of a *Process-Aware Information System* (PAIS): *a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models* [24]. Although not explicitly stated in this definition, it should be noted that the process models mentioned are usually represented in some graphical language, e.g., a Petri-net-like notation. The models are typically instantiated multiple times (e.g., for every customer order) and every instance is handled in a predefined way (possibly with variations).

Classical examples of PAISs are WorkFlow Management (WFM) systems and Business Process Management (BPM) systems. These systems support operational business processes and are driven by an explicit process representation. Given the above definition, one can see that a text editor is not "process aware" insofar as it is used to facilitate the execution of specific tasks without any knowledge of the process of which these tasks are part. A similar comment can be made regarding e-mail clients used to send and receive electronic mail. A task in a process may result in an e-mail being sent, but the e-mail client is unaware of the process it is used in. At any point in time one can send an e-mail to any person without being supported nor restricted by the e-mail client. Text editors and e-mail clients (at least contemporary ones) are applications supporting tasks, not processes. The same applies to a large number of applications used in the context of information systems.

Process awareness is an important property for information systems and the shift from task-driven to PAISs brings a number of advantages [24]:

– The use of explicit process models provides a means for communication between people.
– Systems driven by models rather than code have less problems dealing with change, i.e., if an information system is driven by process models, only the models need to be changed to support evolving or emerging business processes.
– The explicit representation of the processes supported by an organization allows their automated enactment. This may lead to a better performance.
– The explicit representation of processes enables management support at the (re)design level, i.e., explicit process models support (re)design efforts.
– The explicit representation of processes also enables management support at the control level. Generic process monitoring and mining facilities provide useful information about the process as it unfolds. This information can be used to improve the control (or even design) of the process.

A detailed introduction PAISs is beyond the scope of this paper. However, to provide an overview of the important issues, we summarize the classification given in [24]. In addition, we refer to the well-known *workflow patterns* [6,58,70].

2.1 Design-Oriented Versus Implementation-Oriented

Figure 1 summarizes the phases of a typical PAIS life-cycle. In the design phase, processes are designed (or re-designed) based on the outputs of a requirements analysis. In the configuration phase, designs are refined into an implementation, typically by configuring a generic infrastructure for a process-aware information system (e.g. a WFM system, a case handing system, or an EAI platform). After configuration, the enactment phase starts: the operational processes are executed using the configured system. In the diagnosis phase, the operational processes are analyzed to identify problems and to find aspects that can be improved.

Different phases of the PAIS life-cycle call for different techniques and types of tools. For example, the focus of traditional WFM systems is on the lower half of the PAIS life-cycle. They are mainly aimed at supporting process configuration and execution and provide little support for the design and diagnosis phase.

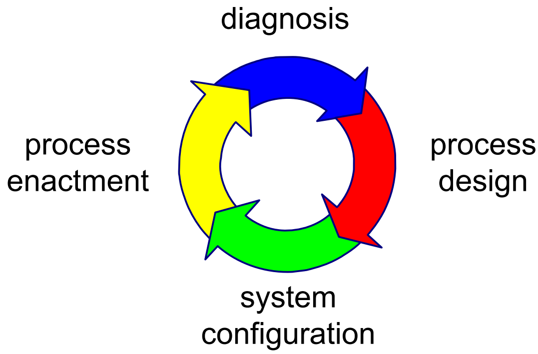


Fig. 1. The PAIS life-cycle [7]

Business process modeling tools are design-oriented and may use all kinds of analysis to evaluate designs. Besides classical analysis techniques such as simulation, more advanced techniques such as process mining come into play, i.e., process improvement by learning from running processes.

2.2 People Versus Software Applications

Another way of classifying PAISs is in terms of the nature of the participants (or resources) they involve, and in particular whether these participants are humans or software applications. In this respect, PAISs can be classified into human-oriented and system-oriented [28] or more precisely into *Person-to-Person* (P2P), *Person-to-Application* (P2A) and *Application-to-Application* (A2A) processes [24].

In P2P processes the participants involved are primarily people, i.e. the processes primarily involve tasks which require human intervention. Job tracking, project management, and groupware tools are designed to support P2P processes. Indeed, the processes supported by these tools usually do not involve entirely automated tasks carried out by applications. Also, the applications that participate in these processes (e.g. project tracking servers, e-mail clients, video-conferencing tools, etc.) are primarily oriented towards supporting computer-mediated interactions.

At the other end of the spectrum, A2A processes are those that only involve tasks performed by software systems. Such processes are typical in the area of distributed computing, and in particular distributed application integration. Transaction processing systems, EAI platforms, and Web-based integration servers are designed to support A2A processes.

P2A processes are those that involve both human tasks and interactions between people, and tasks and interactions involving applications which act without human intervention. Workflow systems fall in the P2A category since they primarily aim at making people and applications work in an integrated manner.

Note that the boundaries between P2P, P2A, and A2A are not crisp. Instead, there is a continuum of techniques and tools from P2P (i.e. manual, human-driven) to A2A (automated, application-driven).

2.3 Predictability of Processes

The degree of structure of the process to be automated (which is strongly linked to its predictability) is frequently used as a dimension to classify PAISs [28]. Structured processes are easier to support than unstructured processes. Moreover, it is also obvious that smaller processes are easier to support than larger ones. However, like in [13,24] we would like to elaborate on the predictability aspect. As Figure 2 shows, we distinguish between *unframed*, *ad hoc framed*, *loosely framed*, and *tightly framed* processes.

A process is said to be *unframed* if there is no explicit process model associated with it. This is the case for collaborative processes supported by groupware systems that do not offer the possibility of defining process models.

A process is said to be *ad hoc framed* if a process model is defined a priori but only executed once or a small number of times before being discarded or changed. This is the case in project management environments where a process model (i.e. a project chart) is often only executed once. It is also the case in grid computing environments, where a scientist may define a process model corresponding to a computation involving a number of datasets and computing resources, and then run this process only once.

A *loosely framed* process is one for which there is an a-priori defined process model and a set of constraints, such that the predefined model describes the "normal way of doing things" while allowing the actual executions of the process to deviate from this model within certain limits.

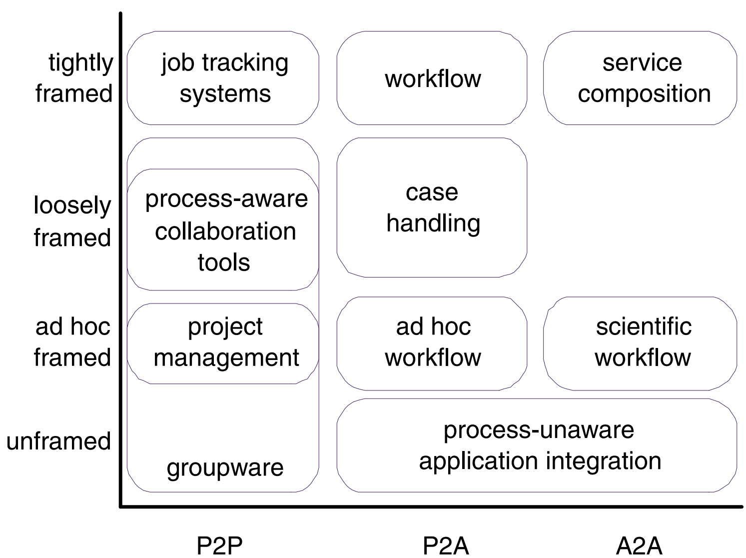


Fig. 2. Type of PAISs and associated development tools [24]

Finally, a *tightly framed* process is one which consistently follows an a-priori defined process model. This is the case of traditional workflow systems.

Figure 2 plots different types of PAISs and PAIS-related tools with respect the degree of framing of the underlying processes (unframed, ad hoc, loosely, or tightly framed), and the nature of the process participants (P2P, P2A, and A2A) [24].

As with P2P, P2A, and A2A processes, the boundaries between unframed, ad hoc framed, loosely framed, and tightly framed processes are not crisp. In particular, there is a continuum between loosely and tightly framed processes. For instance, during its operational life a process considered to be tightly framed can start deviating from its model so often and so unpredictably, that at some point in time it may be considered to have become loosely framed. Conversely, after a large number of cases of a loosely framed process have been executed, a common structure may become apparent, which may then be used to frame the process in a tighter way.

The topic of flexibility in PAISs attracted a lot of attention in the scientific community. Numerous researchers proposed ways of dealing with flexibility and change. Unfortunately, few of these ideas have been adopted by commercial parties. Moreover, it has become clear that there is no "one size fits all" solution, i.e., depending on the application, different types of flexibility are needed. In [60] a taxonomy is given where four types of flexibility are distinguished: (1) *flexibility by design*, (2) *flexibility by deviation*, (3) *flexibility by underspecification*, and (4) *flexibility by change* (both at type and instance levels). This taxonomy shows that different types of flexibility exist. Moreover, different paradigms may be used, i.e., even within one flexibility type there may be different mechanisms that realize different forms of flexibility [63]. All of these approaches aim to support ad hoc framed and/or loosely framed processes.

2.4 Intra-organizational Versus Inter-organizational

Initially, PAISs were mainly oriented towards intra-organizational settings. Focus was on the use of process support technologies (e.g. workflow systems) to automate operational processes involving people and applications inside an organization (or even within an organizational unit). Over the last few years, there has been a push towards processes that cross organizational barriers. Such inter-organizational processes can be one-to-one (i.e. bilateral relations), one-to-many (i.e. an organization interacting with several others) or many-to-many (i.e. a number of partners interacting with each other to achieve a common goal).

The trend towards inter-organizational PAISs is marked by the adoption of SOA and the emergence of web services standards such as BPEL et al.

3 Role of Models

In the previous section, we introduced PAISs and provided a classification. In this section, we focus more on the role of *models*. First of all, we elaborate on the different purposes of models (to provide insights, for analysis purposes, or for enactment). Second, we discuss differences between formal and informal models. Finally, we differentiate between man-made and derived models.

3.1 Purpose

Models can serve different purposes. In fact, the same model can be used for different objectives in the context of a PAIS.

Insight. When developing or improving a PAIS it is important that the different stakeholders get insight into the processes at hand and the way that these processes can or should be supported. Models can be used to discuss requirements, to support design decisions, and to validate assumptions. Moreover, the modeling process itself typically provides new and valuable insights because the modeler is triggered to make things explicit. It is interesting to use the metaphor of a construction drawing for a house. Only when people are confronted with concrete drawings they are able to generate requirements and make their wishes explicit. This holds for houses but also for other complex artifacts such as information systems.

Analysis. Using the metaphor of a construction drawing for a house, it is clear that models can be used to do analysis (e.g., calculating sizes, strengths, etc.). Depending on the type of model, particular types of analysis are possible or not. Moreover, in the context of a PAIS, analysis may focus on the business processes or on the information system itself. For example, the performance of a system (e.g., response times) is not the same as the performance of the processes it supports. Traditionally, most techniques used for the analysis of business processes originate from operations research. Students taking courses in operations management will learn to apply techniques such as simulation, queueing theory, and Markovian analysis. The focus mainly is on *performance analysis* and less attention is paid to the correctness of models. However, *verification* is needed to check whether the resulting system is free of logical errors. Many process designs suffer from deadlocks and livelocks that could have been detected using verification techniques. Notions such as soundness [1,30] can be used to verify the correctness of systems. Similar notions can be used to check interorganizational processes where deadlocks, etc. are more likely to occur [9,39,42].

Enactment. In the context of a PAIS, models are often used for enactment, i.e., based on a model of the process, the corresponding run-time support is generated. In a WFM system, a model of a process suffices to generate the corresponding system support. In other environments, the set of processes is often hard-coded. For example, although ERP systems like SAP have a workflow engine, most processes are hard-coded into the system and can only be changed by programming or changing configuration parameters. As a result, modifications are either time-consuming (because of substantial programming efforts) or restricted by the set of predefined configuration parameters.

3.2 Formality of Models

Related to the purpose of the model is the degree of formality.

Informal models. Informal models cannot be used for enactment and rigorous analysis. Their main purpose is to provide insight, support discussion, etc. We define a model to be informal if it is impossible to determine whether a particular scenario (i.e., a trace of activities) is possible of not.

Formal models. A formal model is able to tell whether a particular sequence of activities is possible or not. For example, given a Petri net it is possible to determine whether a trace corresponds to a possible firing sequence. Even declarative models may be formal. For example, given a model in Declare [47] or plain LTL or CTL [40], it is possible to check whether a trace is possible or not. Formal models typically allow for obtaining insights, analysis, and enactment. However, they may be more difficult to construct than informal models.

The boundaries between formal and informal models seem well-defined. However, in practice one can see many semi-formal models (e.g., BPMN, UML activity diagrams, EPCs, etc.). These models started out as informal models without any formal semantics. However, during the process, subsets have been formalized and are supported by tools that assume particular semantics. The problem is that some people interpret these models in a "formal way" while others use these notations in a rather loose manner. Consider for example the EPC models in SAP where at least 20 percent has serious flaws when one attempts to interpret them in a somewhat unambiguous manner [44]. Besides the differences in interpretation there is the problem that some of the informal concepts create conundrums. For example, the informal semantics of OR-join in EPCs and other languages creates the so-called "vicious cycle" paradox [2,34].

Figure 3 illustrates the relation between industry-driven languages, formal (science-driven) models, and analysis models. The industry-driven languages can be split into informal, semi-formal, and executable. Notations such as BPMN, EPCs, etc. can be seen as semi-formal, i.e., subsets can be interpreted in a precise manner. Languages like BPEL and many other workflow languages are executable because they are supported by concrete workflow engines. Note that these can be considered as formal although there may be different interpretations among different systems. However, in the context of a single execution engine, it is clear what traces are possible and what traces are not possible.

Languages like Petri nets and various process algebraic languages (CSP, CCS, ACP, π-calculus, etc.), are formal and mainly driven by the academic community. The focus is on a clear and unambiguous specification of the process and not on a particular analysis technique. However, such formal languages can often be mapped onto dedicated analysis models. For example, a Petri net can be mapped onto an incidence matrix to calculate invariants or onto a coverability graph to decide on reachability or boundedness.

Let us look at some examples bridging the three layers shown in Figure 3. Woflan is able to translate Staffware, COSA, Protos, and WebSphere models into

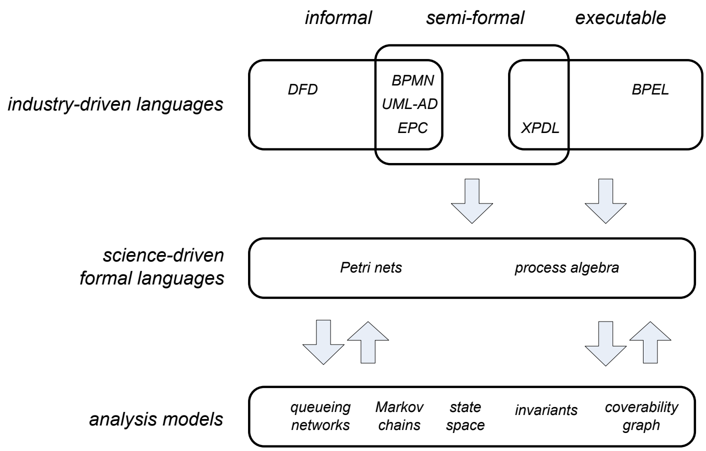


Fig. 3. Relationships among models

Petri nets and then analyze these using the coverability graph and incidence matrix [62]. The toolset BPEL2oWFN/Fiona/LoLA can be used to analyze BPEL models using open workflow nets as an intermediate format [39]. These examples show the possible interplay between various languages.

3.3 Construction Approach

Finally, we distinguish between man-made models and derived models.

Man-made Models. Traditionally, one thinks of models as man-made, i.e., some designer is constructing the model from scratch. When developing a new system or process, this is the only way to obtain models.

Derived Models. If there is already a process or system in place, it is also possible to "derive" models. There are basically two approaches. One approach is to try and reverse engineer models from the system itself, e.g., analyze the code or configuration parameters. Another approach is to extract models based on event logs, i.e., learn from example behavior observed in the past. The next section on process mining will elaborate on the latter type of derived models.

4 Process Mining

After an introduction to PAISs and discussing the various roles of models in the context of such systems, we now focus on a particular analysis technique: *process mining* [12,14,15,19,21,22,23,26,32,35,43,52,64,65]. The reason for elaborating on this particular analysis technique is that our experiences with process mining have dramatically changed our view on PAISs and the role of models in these systems. In fact, the goal of the paper is to *provide a critical reflection on the* *state-of-the-art based on experiences with process mining.* Therefore, we first provide a short introduction to process mining and then elaborate on the lessons learned.

Today's information systems are recording events in so-called event logs. The goal of process mining is to extract information on the process from these logs, i.e., process mining describes a family of *a-posteriori* analysis techniques exploiting the information recorded in the event logs. Typically, these approaches assume that it is possible to sequentially record events such that each event refers to an activity (i.e., a well-defined step in the process) and is related to a particular case (i.e., a process instance). Furthermore, some mining techniques use additional information such as the performer or originator of the event (i.e., the person/resource executing or initiating the activity), the timestamp of the event, or data elements recorded with the event (e.g., the size of an order).

Process mining addresses the problem that most organizations have very limited information about what is actually happening in their organization. In practice, there is often a significant gap between what is prescribed or supposed to happen, and what *actually* happens. Only a concise assessment of the organizational reality, which process mining strives to deliver, can help in verifying process models, and ultimately be used in a process redesign effort or PAIS implementation.

The idea of process mining is to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs. We consider three basic types of process mining (Figure 4).

Discovery. There is no a-priori model, i.e., based on an event log some model is constructed. For example, using the α-algorithm [15] a Petri net can be discovered based on low-level events. Many algorithms have been proposed to discover the control-flow [12,14,15,19,21,22,23,32,35,43,64,65] and few have been prosed to discover other aspects such as the social network [11].

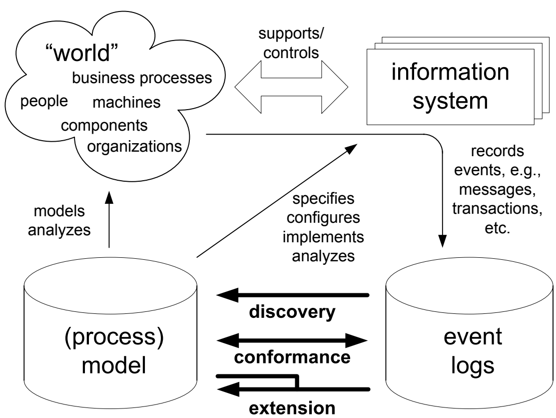


Fig. 4. Three types of process mining: (1) Discovery, (2) Conformance, and (3) Extension

Conformance. There is an a-priori model. This model is used to check if reality, as recorded in the log, conforms to the model and vice versa. For example, there may be a process model indicating that purchase orders of more than one million Euro require two checks. Another example is the checking of the four-eyes principle. Conformance checking may be used to detect deviations, to locate and explain these deviations, and to measure the severity of these deviations. For examples, we refer to the conformance checking algorithms described in [54].

Extension. There is an a-priori model. This model is extended with a new aspect or perspective, i.e., the goal is not to check conformance but to enrich the model. An example is the extension of a process model with performance data, i.e., some a-priori process model is used on which bottlenecks are projected. Another example is the decision mining algorithm described in [53] that extends a given process model with conditions for each decision.

Today, process mining tools are becoming available and are being integrated into larger systems. The ProM framework [3] provides an extensive set of analysis techniques which can be applied to real process enactments while covering the whole spectrum depicted in Figure 4. ARIS PPM was one of the first commercial tools to offer some support for process mining. Using ARIS PPM, one can extract performance information and social networks. Also some primitive form of process discovery is supported. However, ARIS PPM still requires some a-priori modeling. The BPM|suite of Pallas Athena was the first commercial tool to support process discovery without a-priori modeling. Although the above tools have been applied to real-life processes, it remains a challenge to extract suitable process models from event logs. This is illustrated by recent literature [12,14,15,19,21,22,23,32,35,43,64,65].

5 Lessons Learned

Now we would like to provide a critical reflection on the state-of-the-art in PAISs based on our experiences with process mining. The insights provided by process mining are very valuable for the development of the next generation PAISs because they clearly show a mismatch between the models proposed for driving these systems and reality. On the one hand, models tend to oversimplify things resulting in systems that are too restrictive. On the other hand, models fail to capture important aspects of business processes.

In the remainder we present some of the main lessons learned through our various process mining projects.

5.1 Models Do Not Reflect Reality

The first, and probably the most important, lesson is that models typically provide a very naive view of reality. Reality is typically much more dynamic and complex than what is captured in models. Models should abstract from details and aspects not relevant for the purpose of the model. However, the discrepancies

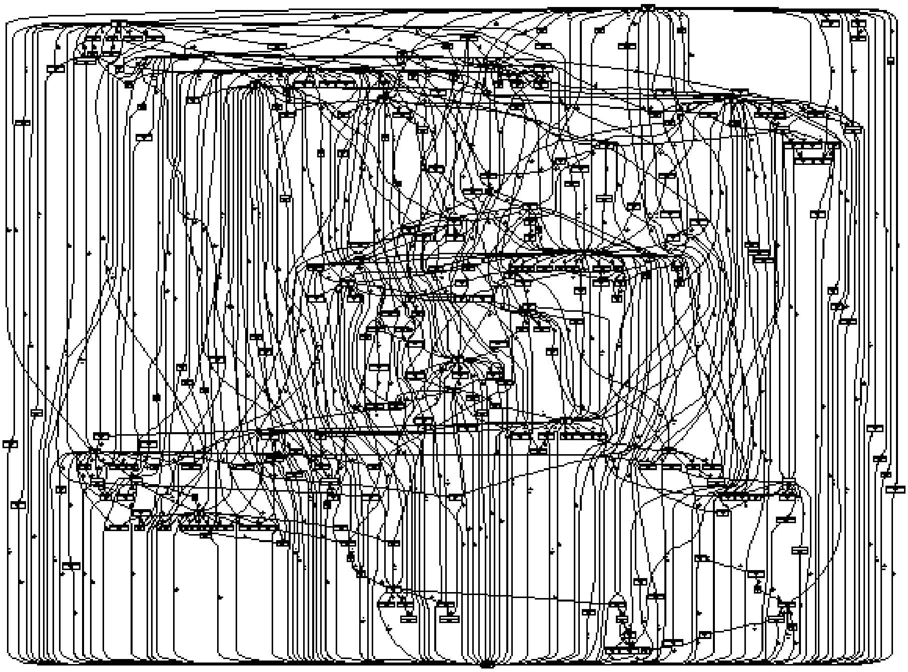


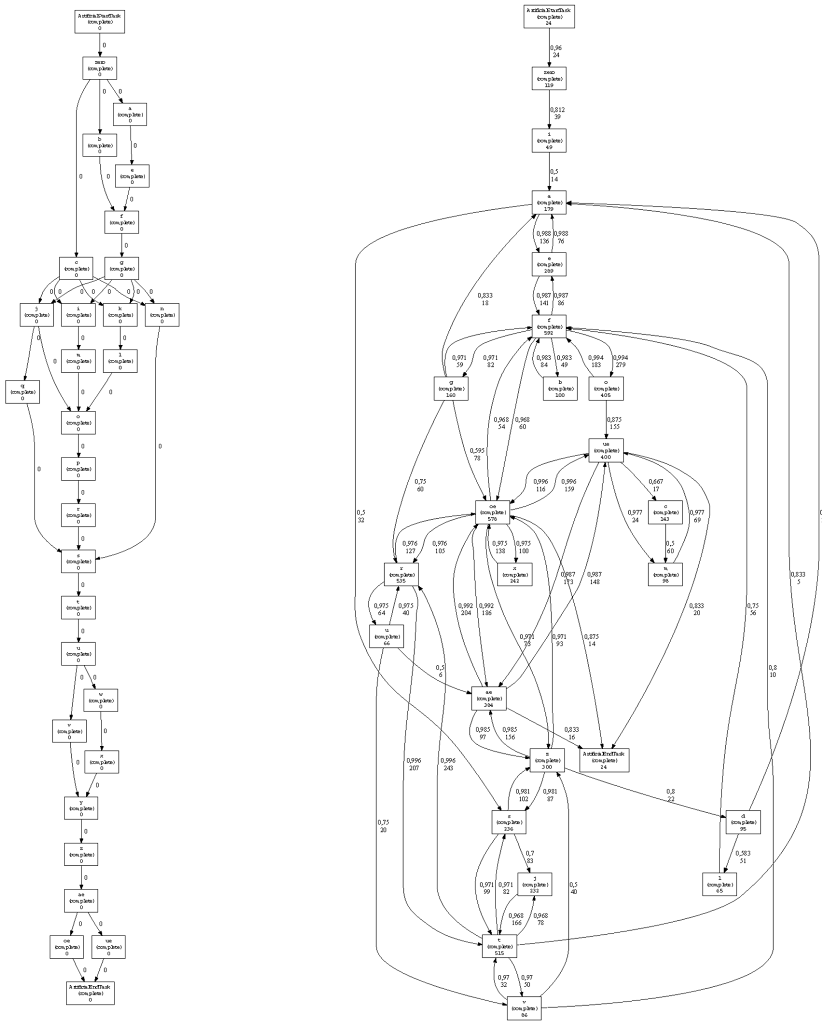
Fig. 5. Process discovered based on an event log with information about 2712 patients

that can be found between models and reality can typically not be justified by reasons of abstraction.

To illustrate this consider the process model shown in Figure 5. This model was discovered using ProM's Heuristics Miner [64] based on the data of 2712 patients treated in a Dutch hospital. The log contained 29258 events (i.e., +/- 10.8 events per case) corresponding to 264 activities. The discovered process model reflects the complexity of care processes. One may expect such "spaghetti-like processes" in a hospital. However, we have found similarly unstructured processes in many environments where one would expect more structured processes (e.g., municipalities, banks, insurance companies, etc.). It is important to note that the spaghetti-like process shown in Figure 5 is not due to limitations of the process mining techniques used, i.e., it is completely caused by the real complexity of the process.

Insights provided by process models such as the one shown in Figure 5 serve as a reality check for any PAIS implementation. Without a complete understanding of the processes at hand, the PAIS is destined to fail.

To illustrate the discrepancies between models and reality further, consider Figure 6 taken from [55]. These models have been obtained when analyzing one of the test processes of ASML (the leading manufacturer of wafer scanners in the world). ASML designs, develops, integrates and services advanced systems to produce semiconductors. In short, it makes the wafer scanners that print the chips. These wafer scanners are used to manufacture semi-conductors



(a) Reference process model (b) Discovered process model

Fig. 6. Two heuristic nets [43,64] showing the difference between (a) the translated reference model for the test process on job-step level and (b) the discovered process model based on log which was mapped onto the job-step level [55]

(e.g., processors in devices ranging from mobile phones ad MP3 players to desktop computers). At any point in time, ASML's wafer scanners record events that can easily be distributed over the internet. Hence, any event that takes place during the test process is recorded. The availability of these event logs and the desire of ASML to improve the testing process, triggered the case study reported in [55]. If we apply ProM's discovery to the low-level logs, we obtain

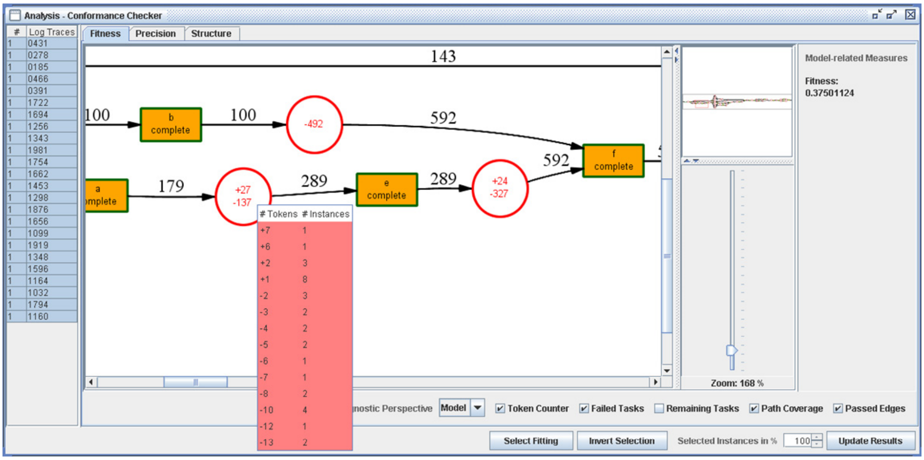


Fig. 7. Screenshot of ProM's Conformance Checker while analyzing the difference between the reference model and reality [55]

a spaghetti-like process similar to the one shown in Figure 5. However, using domain knowledge the low level log can be translated to an event log at the so-called job-step level. ASML also provided us with a reference model at the job-step level. This model was used to instruct the test engineers. Figure 6(a) shows the reference model. The discovered model is shown in Figure 6(b). It is interesting to note that the discovered model allows for much more scenarios than the reference model.

In Figure 7 we used ProM's Conformance Checker while analyzing the deviations in ASML's test process. As shown the fitness is only 37.5 percent, i.e., roughly one third of the events can be explained by the model indicating that "exceptions are the rule" [54]. By looking at the most frequent paths that appear in Figure 6(b) and not in Figure 6(a) and at the diagnostics provided in Figure 7 it is possible to pinpoint the most important deviations. Note that deviations are not necessarily a bad thing and may reflect (desirable) flexibility. We will elaborate on this in Section 5.3.

The results presented in this section are not exceptional, i.e., many processes turn out to be more spaghetti-like than expected. Nevertheless, most attention in both academia and industry is given to the analysis and use of models and not to the way to obtain them. Both sides take models as a starting point. Analysis techniques, process engines, etc. focus on what to do with models rather than obtaining faithful models. Therefore, we would like to stress the need for more emphasis on the faithfulness of models. For example, analysis results are only meaningful if the corresponding models are adequate.

5.2 A Human's Characteristics Are Difficult to Capture

In the previous section, we focused in discrepancies between the control-flow as modeled and the real control-flow. When it comes to resources similar problems emerge, especially if the resources are human. This mismatch becomes evident when comparing the behavior of humans observed when using process mining techniques and the behavior of humans assumed in simulation tools [10]. In the remainder, we focus on the problems encountered when modeling people for simulation purposes. However, the insights also apply to other analysis methods and enactment support (e.g., software for work distribution).

In practice there are few people that only perform activities for a single process. Often people are involved in many different processes, e.g., a manager, doctor, or specialist may perform tasks in a wide range of processes. However, simulation often focuses on a single process. Suppose a manager is involved in 10 different processes and spends about 20 percent of his time on the process that we want to analyze. In most simulation tools it is impossible to model that a resource is only available 20 percent of the time. Hence, one needs to assume that the manager is there all the time and has a very low utilization. As a result the simulation results are too optimistic. In the more advanced simulation tools, one can indicate that resources are there at certain times in the week (e.g., only on Monday). This is also an incorrect abstraction as the manager distributes his work over the various processes based on priorities and workload. Suppose that there are 5 managers all working 20 percent of their time on the process of interest. One could think that these 5 managers could be replaced by a single manager (5*20%=1*100%). However, from a simulation point of view this is an incorrect abstraction. There may be times that all 5 managers are available and there may be times that none of them are available.

Another problem is that people work at different speeds based on their workload, i.e., it is not just the distribution of attention over various processes, but also a person's absolute working speed influences his/her capacity for a particular process. There are various studies that suggest a relation between workload and performance of people. A well-known example is the so-called Yerkes-Dodson law [69]. The Yerkes-Dodson law models the relationship between arousal and performance as an inverse U-shaped curve. This implies that for a given individual and a given type of tasks, there exists an optimal arousal level. This is the level where the performance has its maximal value. Thus work pressure is productive, up to a certain point, beyond which performance collapses. Although this phenomenon can be easily observed in daily life, today's business process simulation tools do not support the modeling of workload dependent processing times.

As indicated earlier, people may be involved in different processes. Moreover, they may work part-time (e.g., only in the morning). In addition to their limited availabilities, people have a tendency to work in batches (cf. Resource Pattern 38: Piled Execution [58]). In any operational process, the same task typically needs to be executed for many different cases (process instances). Often people prefer to let work-items related to the same task accumulate, and then process all of these in one batch. In most simulation tools a resource is either available or not, i.e., it is assumed that a resource is eagerly waiting for work and immediately reacts to any work-item that arrives. Clearly, this does not do justice to the way people work in reality. For example, consider how and when people reply to e-mails. Some people handle e-mails one-by-one when they arrive while others process all of their e-mails at fixed times in batch.

Also related is the fact that calendars and shifts are typically ignored in simulation tools. While holidays, lunch breaks, etc. can heavily impact the performance of a process, they are typically not incorporated in the simulation model.

All these observations show that it is very difficult to adequately capture human activity in simulation models. As a result it is not uncommon that the simulation model predicts a flow time of hours while in reality the average flow time is weeks. In [10] the effects of some of these incorrect assumptions on the simulation results are shown. Using process mining one can get insight into the way that humans actually work and use this to build more faithful simulation models.

Note that the difficulties encountered when characterizing humans is not only relevant for simulation but also for enactment support. It can be observed that only few of the resource patterns [58] are supported by contemporary PAISs. Moreover, insights such as the Yerkes-Dodson law are not used by today's PAISs and systems are unable to predict problems. The lack of understanding and limited functionality impairs the successfulness of PAISs.

5.3 Spaghetti and Flexibility: Two Sides of the Same Coin

The topic of flexibility in the context WFM systems has been addressed by many authors [16,18,25,27,47,48,51,66]. See the taxonomy in [60] or the flexibility patterns in [63] to get an overview of the different approaches proposed in literature. See also [8,17,28,31,46,50,59] for other classifications of flexibility. Researchers proposed numerous ways of dealing with flexibility and change. Unfortunately, few of these ideas have been adopted by commercial parties. Process mining can expose the need for flexibility. Spaghetti-like processes as shown in Figure 5 and the quantification of non-conformance illustrated by Figure 7 illustrate the need for flexibility.

When building a PAIS for existing processes, process mining can be used to identify the flexibility needs. When looking at the spaghetti-like processes discovered using process mining, it becomes evident that one has to decide on what kinds of variability are actually desired. Some deviations are good because they correspond to adequate responses to requests from the environment. Other deviations may be undesirable because they impair quality or efficiency.

It is easy to say that PAISs should provide more flexibility. However, process mining also shows that it is very difficult to actually do this. It seems that much of the research in this domain is rather naive. For example, it is ridiculous to assume that end-users will be able construct or even understand process models such as the one depicted in Figure 5.

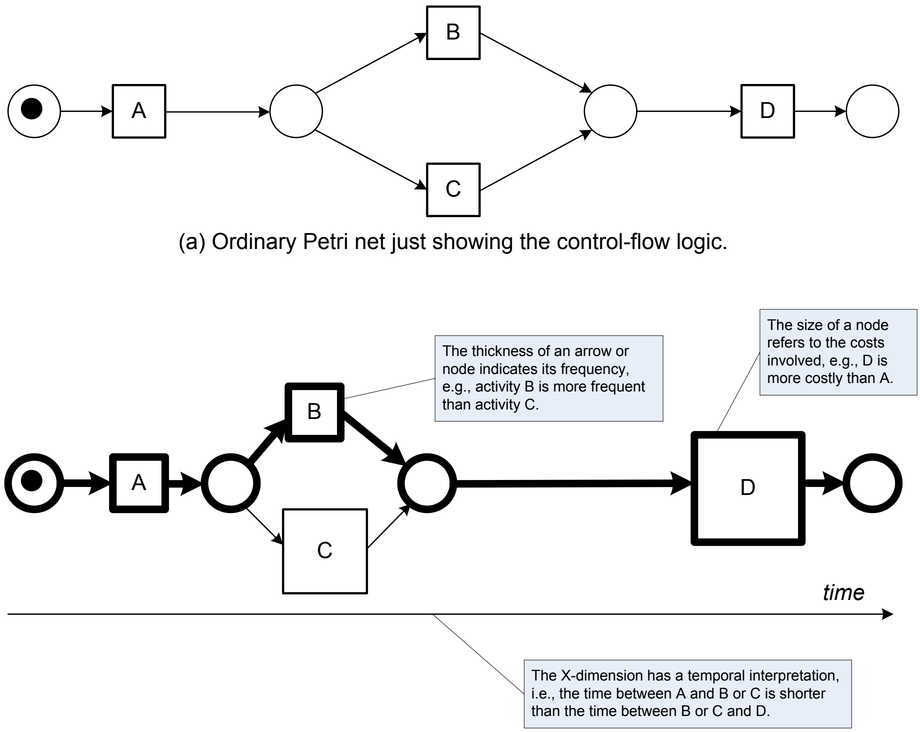
5.4 Process Models Should Be Treated as Maps

There are dozens of process modeling languages. Most of these languages provide some graphical notation with split and join nodes of particular types (AND, XOR, etc.). Although there are important semantical differences between these notations, the basic idea is always to draw a graph describing the routing of process instances (cases). This seems to be a good approach as it is adopted by most vendors and common practice in industry. Nevertheless, our experiences with process mining have revealed several weaknesses associated with this classical approach. Diagrams like Figure 5 show that automatically derived models are difficult to understand and lack expressiveness. This triggered us to look at process models as ordinary geographic maps. The "map metaphor" reveals some interesting insights.

- There is no such thing as "the map". One may use city maps, highway maps, hiking maps, cycling maps, booting maps, etc. depending on the purpose for which it is intended to be used. All of these maps refer to the same reality. However, nobody would aim at trying to construct a single map that suits all purposes. Unfortunately, when it comes to processes one typically aims at a single map.
- Another insight is that process models do not exploit colors, dimensions, sizes, etc. It is remarkable that process models typically have shapes (nodes and arcs) of a fixed size, and, even if the size is variable, it has no semantical interpretation. Colors in process models are only used for beatification and not to express things like intensity, costs, etc. On a geographic map the X and Y dimension have a clear interpretation. These dimensions are not explicitly used when drawing process models.

To illustrate the above, consider Figure 8 showing two times the same Petri net. Although from a logical point of view the Petri nets are identical, the lower one also shows frequencies, costs, and time. For example, it is shown that the path (A, B, D) is much more frequent than the path (A, C, D). Moreover, activities C and D are more costly than A and B. The X-dimension is used to reflect time. The horizontal position corresponds to the average time at which activity takes places after the model is initiated by placing a token on the input place. It clearly shows that most time is spent waiting for the execution of D. Figure 8(b) is still very primitive compared to the drawing of maps. Maps typically also use colors and other intuitive annotations to indicate relevant information. Moreover, maps abstract and aggregate. Abstraction is used to leave out things that are less significant (i.e., dirt roads and small townships). Aggregation is used to take things together. For example, the roads of a city are taken together into a single shape. In terms of Figure 8, abstraction could mean that C is removed because it is too insignificant. Aggregation should be used to group A, B, and C into a single node because they typically occur together in a short time distance.

Today, electronic maps overcome some of the limitations of paper maps. As indicated above one may use different maps (city maps, highway maps, etc.) depending on the purpose. When using Google Maps or a car navigation system like TomTom it is possible to dynamically zoom-in, zoom-out, change focus, or change the type of information. Moreover, these electronic maps are increasingly used to project dynamic information on. For example TomTom is able to show



(a) Ordinary Petri net just showing the control-flow logic.

(b) Petri net also showing other dimensions (frequency, time, and costs).

Fig. 8. Using the map metaphor for drawing process models

traffic jams, fuel stations with the lowest prices, weather information, etc. These ideas can also be used for process models.

– Process models should allow for different views, i.e., it should be possible to zoom-in and zoom-out seamlessly. The static decompositions used by contemporary drawing tools force the user to view processes in a fixed manner. Moreover, decompositions typically address the needs of a technical designer rather than an end-user. Hence the challenge is to be able to support easy navigation and seamlessly zooming-in/out when viewing process models.
– It should be possible to project dynamic information on top of process models. For example, it should be possible to view current process instances projected on the process model and to animate history by replaying past events on the same process model. This is similar to showing real-time traffic information by a car navigation system like TomTom.

The limitations related to the representation and visualization of process models mentioned above became evident based on experiences gathered in many process mining projects. It seems that the "map metaphor" can be used to present process models and process information in completely new ways [29,37]. Few

Fig. 9. ProM's Fuzzy Miner implements some of the ideas learned from (electronic) maps [29]

researchers have been investigating such ideas. Here we would like to point out two ideas we have been working on. In the context of YAWL [5,37,61,72], we showed that it is possible to show current work items on top of various maps. Work items can be shown on top of a geographic map, a process model, a time chart, an organizational model, etc. In the context of ProM, we have used the "map metaphor" to enhance the so-called Fuzzy Miner [29]. As presented in [29], four ideas are being combined in ProM's Fuzzy Miner to draw maps of process models.

- *Aggregation:* To limit the number of information items displayed, maps often show coherent clusters of low-level detail information in an aggregated manner. One example are cities in road maps, where particular houses and streets are combined within the citys transitive closure (e.g., the city of Eindhoven in Figure 9).
- *Abstraction:* Lower-level information which is insignificant in the chosen context is simply omitted from the visualization. Examples are bicycle paths, which are of no interest in a motorists map.
- *Emphasis:* More significant information is highlighted by visual means such as color, contrast, saturation, and size. For example, maps emphasize more important roads by displaying them as thicker, more colorful and contrasting lines (e.g., motorway "E25" in Figure 9).
- *Customization:* There is no one single map for the world. Maps are specialized on a defined local context, have a specific level of detail (city maps vs highway maps), and a dedicated purpose (interregional travel vs alpine hiking).

Figure 10 shows screenshot of ProM's Fuzzy Miner [29]. The left screen shows a discovered model. Note that the thickness of each arc is determined by the number of times this path is taken (i.e., frequency). Moreover, some nodes and arcs have been left out because they are insignificant. The left screen shows an animation based on historic information. This animation shows the actual execution of cases on top of the discovered model.

It is obvious that the ideas presented here are not limited to process mining. When developing, analyzing, or controlling a PAIS, such visualizations can be very useful.

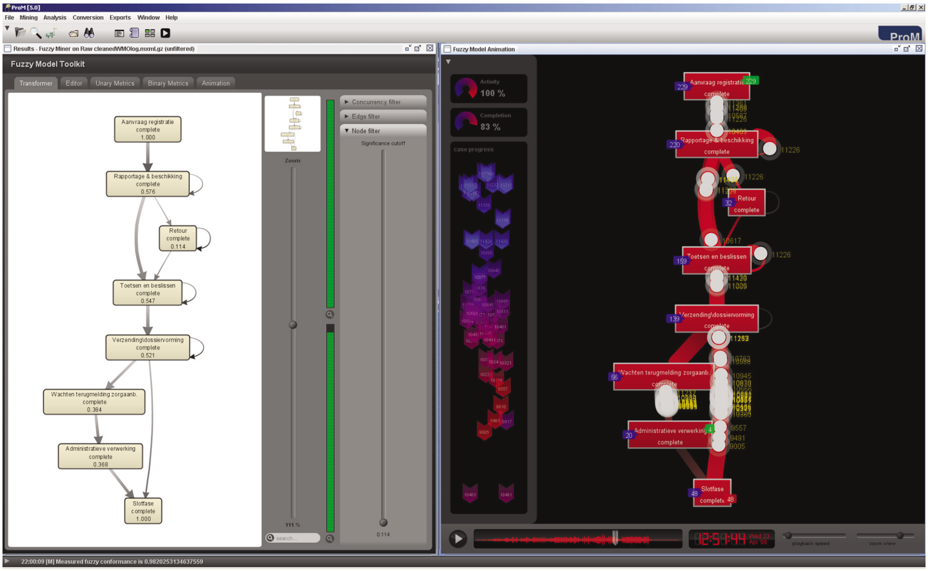


Fig. 10. ProM's Fuzzy Miner (left) and the corresponding animation facility (right)

5.5 Analysis Techniques Do Not Use the Information Available

The last lesson to be learned is related to the limited use of existing artifacts. People tend to model things from scratch and do not use information that is already recorded in information systems. In practice, it is time consuming to construct a good process model. For example, when constructing a simulation model one not only has to construct a model but also determine the input parameters. A pitfall of current approaches is that existing artifacts (models, logs, data, etc.) are not used in a direct and systematic manner. If a PAIS is used, there are often models that are used to configure the system (e.g., workflow schemas). Today, these models are typically disconnected from the simulation models and created separately. Sometimes a business process modeling tool is used to make an initial process design. This design can be used for simulation purposes when using a tool like Protos or ARIS. When the designed process is implemented, another system is used and the connection between the implementation model and the design model is lost. It may be that at a later stage, when the process needs to be analyzed, a simulation model is built from scratch. This is a pity as the PAIS contains most of the information required. As a result the process is "reinvented" again and again, thus introducing errors and unnecessary work. The lack of reuse also applies to other sources of information. For example, the PAIS may provide detailed event logs. Therefore, there is no need to "invent" processing times, arrival times, and routing probabilities, etc. All of this information can be extracted from the logs. Note that a wealth of information can be derived from event logs. In fact, in [56] it is demonstrated that complete simulation models can be extracted from event logs.

Contemporary simulation tools tend to support experiments that start in an arbitrary initial state (without any cases in the pipeline) and then simulate the process for a long period to make statements about the steady-state behavior. However, this steady-state behavior does not exist (the environment of the process changes continuously) and is thus considered irrelevant by the manager. Moreover, the really interesting questions are related to the near future. Therefore, it seems vital to also support transient analysis, often referred to as *short-term simulation* [49,57,71]. The "fast-forward button" provided by short-term simulation is a useful option, however, it requires the use of the current state. Fortunately, when using a PAIS it is relatively easy to obtain the current state and load this into the simulation model.

The above not only applies to simulation models. Also other types of analysis can benefit from the information stored in and recorded by the PAIS [57].

6 Conclusion

Workflow management systems, case-handling systems, enterprise information systems, etc. are all examples of PAISs. We introduced these systems by characterizing them in several ways. Moreover, we elaborated on the role of process models in the context of such systems. After this introduction, we focused on lessons learned from process mining. The goal of process mining is to extract information from event logs. These event logs can be used to automatically generate models (process discovery) or to compare models with reality (conformance checking).

Extensive experience gathered through various process mining projects, has revealed important lessons for the development and use of PAISs. The first lesson is that models typically provide a very naive view of reality. The second lesson is that it is far from trivial to adequately capture the characteristics of human actors. The third lesson is that the true need for flexibility can be seen by analyzing spaghetti-like process models. The fourth lesson is that the way we view processes can be improved dramatically by using the "map metaphor". The fifth lesson is that many artifacts (models and logs) remain unused by today's analysis approaches.

Although these lessons were triggered by the application of process mining to many real-life logs, they are useful for the whole PAIS life-cycle. It does not make any sense to talk about analysis or enactment, without a good and deep understanding of the processes at hand.

Acknowledgements

Parts of the work reported in this paper are the result of joint work with many other researchers from TU/e and QUT. Section 2 is based on joint work with Marlon Dumas and Arthur ter Hofstede [24]. The experiences reported in Section 5 were only possible because the maturity of our process mining tool ProM. Therefore, the author would like to thank the many people involved in the development of ProM.

References

1. van der Aalst, W.M.P.: The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers 8(1), 21–66 (1998)
2. van der Aalst, W.M.P., Desel, J., Kindler, E.: On the Semantics of EPCs: A Vicious Circle. In: Nüttgens, M., Rump, F.J. (eds.) Proceedings of the EPK 2002: Business Process Management using EPCs, Trier, Germany, November 2002, pp. 71–80. Gesellschaft für Informatik, Bonn (2002)
3. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Mans, R.S., de Medeiros, A.K.A., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W., Weijters, A.J.M.M.: ProM 4.0: Comprehensive Support for Real Process Analysis. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 484–494. Springer, Heidelberg (2007)
4. van der Aalst, W.M.P., van Hee, K.M.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge (2004)
5. van der Aalst, W.M.P., ter Hofstede, A.H.M.: YAWL: Yet Another Workflow Language. Information Systems 30(4), 245–275 (2005)
6. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14(1), 5–51 (2003)
7. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
8. van der Aalst, W.M.P., Jablonski, S.: Dealing with Workflow Change: Identification of Issues and Solutions. International Journal of Computer Systems, Science, and Engineering 15(5), 267–276 (2000)
9. van der Aalst, W.M.P., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K.: From Public Views to Private Views: Correctness-by-Design for Services. In: Dumas, M., Heckel, R. (eds.) WS-FM 2007. LNCS, vol. 4937, pp. 139–153. Springer, Heidelberg (2008)
10. van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russell, N.: Business Process Simulation: How to get it right? In: vom Brocke, J., Rosemann, M. (eds.) International Handbook on Business Process Management. Springer, Berlin (2008)
11. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering Social Networks from Event Logs. Computer Supported Cooperative work 14(6), 549–593 (2005)
12. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., Alves de Medeiros, A.K., Song, M., Verbeek, H.M.W.: Business Process Mining: An Industrial Application. Information Systems 32(5), 713–732 (2007)
13. van der Aalst, W.M.P., Stoffele, M., Wamelink, J.W.F.: Case Handling in Construction. Automation in Construction 12(3), 303–320 (2003)
14. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering 47(2), 237–267 (2003)
15. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1128–1142 (2004)
16. van der Aalst, W.M.P., Weske, M., Grünbauer, D.: Case Handling: A New Paradigm for Business Process Support. Data and Knowledge Engineering 53(2), 129–162 (2005)
17. Adams, M.: Facilitating Dynamic Flexibility and Exception Handling for Workflows. Phd thesis, Queensland University of Technology (2007)

18. Adams, M., ter Hofstede, A.H.M., van der Aalst, W.M.P., Edmond, D.: Dynamic, Extensible and Context-Aware Exception Handling for Workflows. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 95–112. Springer, Heidelberg (2007)
19. Agrawal, R., Gunopulos, D., Leymann, F.: Mining Process Models from Workflow Logs. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 469–483. Springer, Heidelberg (1998)
20. Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guízar, A., Kartha, N., Liu, C.K., Khalaf, R., Koenig, D., Marin, M., Mehta, V., Thatte, S., Rijn, D., Yendluri, P., Yiu, A.: Web Services Business Process Execution Language Version 2.0 (OASIS Standard). WS-BPEL TC OASIS (2007), http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html
21. Bergenthum, R., Desel, J., Lorenz, R., Mauser, S.: Process Mining Based on Regions of Languages. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 375–383. Springer, Heidelberg (2007)
22. Datta, A.: Automating the Discovery of As-Is Business Process Models: Probabilistic and Algorithmic Approaches. Information Systems Research 9(3), 275–301 (1998)
23. van Dongen, B.F., van der Aalst, W.M.P.: Multi-Phase Process Mining: Building Instance Graphs. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 362–376. Springer, Heidelberg (2004)
24. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley & Sons, Chichester (2005)
25. Dustdar, S.: Caramba - A Process-Aware Collaboration System Supporting Ad Hoc and Collaborative Processes in Virtual Teams. Distributed and Parallel Databases 15(1), 45–66 (2004)
26. Dustdar, S., Gombotz, R.: Discovering Web Service Workflows Using Web Services Interaction Mining. International Journal of Business Process Integration and Management 1(4), 256–266 (2006)
27. Ellis, C.A., Keddara, K., Rozenberg, G.: Dynamic Change within Workflow Systems. In: Comstock, N., Ellis, C., Kling, R., Mylopoulos, J., Kaplan, S. (eds.) Proceedings of the Conference on Organizational Computing Systems, ACM SIGOIS, Milpitas, California, August 1995, pp. 10–21. ACM Press, New York (1995)
28. Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases 3, 119–153 (1995)
29. Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007)
30. van Hee, K.M., Sidorova, N., Voorhoeve, M.: Generalised Soundness of Workflow Nets Is Decidable. In: Cortadella, J., Reisig, W. (eds.) ICATPN 2004. LNCS, vol. 3099, pp. 197–215. Springer, Heidelberg (2004)
31. Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., Teschke, M.: A Comprehensive Approach to Flexibility in Workflow Management Systems. In: Georgakopoulos, G., Prinz, W., Wolf, A.L. (eds.) Work Activities Coordination and Collaboration (WACC 1999), pp. 79–88. ACM press, San Francisco (1999)
32. Herbst, J.: A Machine Learning Approach to Workflow Management. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS, vol. 1810, pp. 183–194. Springer, Heidelberg (2000)

33. Jablonski, S., Bussler, C.: Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press, London (1996)
34. Kindler, E.: On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. Data and Knowledge Engineering 56(1), 23–40 (2006)
35. Lamma, E., Mello, P., Montali, M., Riguzzi, F., Storari, S.: Inducing Declarative Logic-Based Models from Labeled Traces. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 344–359. Springer, Heidelberg (2007)
36. Lawrence, P. (ed.): Workflow Handbook 1997, Workflow Management Coalition. John Wiley and Sons, New York (1997)
37. de Leoni, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Visual Support for Work Assignment in Process-Aware Information Systems. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 67–83. Springer, Heidelberg (2008)
38. Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques. Prentice-Hall PTR, Upper Saddle River (1999)
39. Lohmann, N., Massuthe, P., Stahl, C., Weinberg, D.: Analyzing Interacting BPEL Processes. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 17–32. Springer, Heidelberg (2006)
40. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer, New York (1991)
41. Marinescu, D.C.: Internet-Based Workflow Management: Towards a Semantic Web. Wiley Series on Parallel and Distributed Computing, vol. 40. Wiley-Interscience, New York (2002)
42. Massuthe, P., Reisig, W., Schmidt, K.: An Operating Guideline Approach to the SOA. Annals of Mathematics, Computing & Teleinformatics 1(3), 35–43 (2005)
43. de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic Process Mining: An Experimental Evaluation. Data Mining and Knowledge Discovery 14(2), 245–304 (2007)
44. Mendling, J., Neumann, G., van der Aalst, W.M.P.: Understanding the Occurrence of Errors in Process Models Based on Metrics. In: Curbera, F., Leymann, F., Weske, M. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 113–130. Springer, Heidelberg (2007)
45. zur Muehlen, M.: Workflow-based Process Controlling: Foundation, Design and Application of workflow-driven Process Information Systems, Logos, Berlin (2004)
46. Pesic, M.: Constraint-based Workflow Management Systems: Shifting Control to Users. Phd thesis, Eindhoven University of Technology (May 2008)
47. Pesic, M., Schonenberg, M.H., Sidorova, N., van der Aalst, W.M.P.: Constraint-Based Workflow Models: Change Made Easy. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 77–94. Springer, Heidelberg (2007)
48. Reichert, M., Dadam, P.: ADEPTflex: Supporting Dynamic Changes of Workflow without Loosing Control. Journal of Intelligent Information Systems 10(2), 93–129 (1998)
49. Reijers, H.A., van der Aalst, W.M.P.: Short-Term Simulation: Bridging the Gap between Operational Control and Strategic Decision Making. In: Hamza, M.H. (ed.) Proceedings of the IASTED International Conference on Modelling and Simulation, pp. 417–421. IASTED/Acta Press, Anaheim (1999)
50. Rinderle, S., Reichert, M., Dadam, P.: Evaluation of Correctness Criteria for Dynamic Workflow Changes. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 41–57. Springer, Heidelberg (2003)

51. Rinderle, S., Reichert, M., Dadam, P.: Correctness Criteria For Dynamic Changes in Workflow Systems: A Survey. Data and Knowledge Engineering 50(1), 9–34 (2004)
52. Rozinat, A., van der Aalst, W.M.P.: Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 163–176. Springer, Heidelberg (2006)
53. Rozinat, A., van der Aalst, W.M.P.: Decision Mining in ProM. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 420–425. Springer, Heidelberg (2006)
54. Rozinat, A., van der Aalst, W.M.P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems 33(1), 64–95 (2008)
55. Rozinat, A., de Jong, I.S.M., Günther, C.W., van der Aalst, W.M.P.: Process Mining of Test Processes: A Case Study. In: BETA Working Paper Series, WP 220. Eindhoven University of Technology, Eindhoven (2007)
56. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering Colored Petri Nets From Event Logs. International Journal on Software Tools for Technology Transfer 10(1), 57–74 (2008)
57. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.: Workflow Simulation for Operational Decision Support Using Design, Historic and State Information. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 196–211. Springer, Heidelberg (2008)
58. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow Resource Patterns: Identification, Representation and Tool Support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
59. Sadiq, S., Sadiq, W., Orlowska, M.: Pockets of Flexibility in Workflow Specification. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 513–526. Springer, Heidelberg (2001)
60. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., van der Aalst, W.M.P.: Process Flexibility: A Survey of Contemporary Approaches. In: Dietz, J., Albani, A., Barjis, J. (eds.) Advances in Enterprise Engineering I. LNBIP, vol. 10, pp. 16–30. Springer, Heidelberg (2008)
61. Streit, A., Pham, B., Brown, R.: Visualisation Support for Managing Large Business Process Specifications. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 205–219. Springer, Heidelberg (2005)
62. Verbeek, H.M.W., Basten, T., van der Aalst, W.M.P.: Diagnosing Workflow Processes using Woflan. The Computer Journal 44(4), 246–279 (2001)
63. Weber, B., Reichert, M., Rinderle-Ma, S.: Change Patterns and Change Support Features: Enhancing Flexibility in Process-Aware Information Systems. Data and Knowledge Engineering 66(3), 438–466 (2008)
64. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integrated Computer-Aided Engineering 10(2), 151–162 (2003)
65. Wen, L., van der Aalst, W.M.P., Wang, J., Sun, J.: Mining process models with non-free-choice constructs. Data Mining and Knowledge Discovery 15(2), 145–180 (2007)
66. Weske, M.: Formal Foundation and Conceptual Design of Dynamic Adaptations in a Workflow Management System. In: Sprague, R. (ed.) Proceedings of the Thirty-Fourth Annual Hawaii International Conference on System Science (HICSS-34). IEEE Computer Society Press, Los Alamitos (2001)

67. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, Berlin (2007)
68. White, S.A., et al.: Business Process Modeling Notation Specification (Version 1.0, OMG Final Adopted Specification) (2006)
69. Wickens, C.D.: Engineering Psychology and Human Performance. Harper (1992)
70. Workflow Patterns Home Page, http://www.workflowpatterns.com
71. Wynn, M.T., Dumas, M., Fidge, C.J., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Business Process Simulation for Operational Decision Support. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) BPM Workshops 2007. LNCS, vol. 4928, pp. 66–77. Springer, Heidelberg (2008)
72. YAWL Home Page, http://www.citi.qut.edu.au/yawl/