Decision Support Based on 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@tm.tue.nl (telephone: +31 40 247.4295, fax: +31 40 243.2612)

Process mining techniques allow for the analysis of business processes based on event logs. For example, the audit trails of a workflow management system, the transaction logs of an enterprise resource planning system, and the electronic patient records in a hospital can be used to discover models describing processes, organizations, and products. Moreover, such event logs can also be used to compare event logs with some a-priori model to see whether the observed reality conforms to some prescriptive or descriptive model. This chapter takes the MXML format as a starting point, i.e., a format that stores event logs in a unified manner. Based on this format, we will show how process mining techniques can be used to support decision making in business processes.

Keywords: Process Mining, Business Activity Monitoring, Business Process Intelligence, Data Mining.

---

1. Introduction

Process mining techniques [3] can be used in a variety of application domains ranging from manufacturing and e-business to health care and auditing. Unlike many other decision support systems the focus is on the analysis of the current situation rather than evaluating redesigns or proposing improvements. We believe that successful improvements are only possible if one truly understands what is happening in the current business processes. We have experienced that often managers and users do not have a clear view of the *real* process. People tend to think in terms of highly simplified processes and their views on these processes often contain an initial bias. Therefore, it is vital to have an objective understanding of reality. Moreover, it is often not sufficient to understand things at an aggregate level. One needs to take notice of causalities at a lower level, i.e., at the level of individual activities within specific cases rather than at the level of frequencies and averages. For example, a manager in a hospital may know the number of knee operations and the average flow time of patients that have a knee problem. However, the causalities and other subtle decencies between the various steps in the process are often hidden. For example, it may be important to know that some steps are frequently executed in an undesirable order or that for many patients some steps are executed multiple times. The goal of process mining is to provide a variety of views on the processes. This can be done by discovering models based on directly observing reality or by comparing reality with some a-priori model. *The outcome of process mining is a better understanding of the process and accurate models that can safely be used for decision* *support because they reflect reality*. Before we elaborate on process mining we first position this chapter in the broader *BPM (Business Process Management)* context.

Buzzwords such as BAM (Business Activity Monitoring), BOM (Business Operations Management), BPI (Business Process Intelligence) illustrate the interest in closing the BPM loop [1,8]. This is illustrated by Figure 1 which shows the level of support in four different years using the BPM lifecycle. The lifecycle identifies four different phases: *process design* (i.e., making a workflow schema), *system configuration* (i.e., getting a system to support the designed process), *process enactment* (i.e., the actual execution of the process using the system), and *diagnosis* (i.e., extracting knowledge from the process as it has been executed). As Figure 1 illustrates, BPM technology (e.g., workflow management systems but also other process-aware information systems [8]) started with a focus on getting the system to work (i.e., the system configuration phase). Since the early nineties BPM technology matured and more emphasis was put on supporting the process design and process enactment phases in a better way. Now many vendors are trying to close the BPM lifecycle by adding diagnosis functionality. The buzzwords BAM, BOM, BPI, etc. illustrate these attempts [8,9,13,14,17].

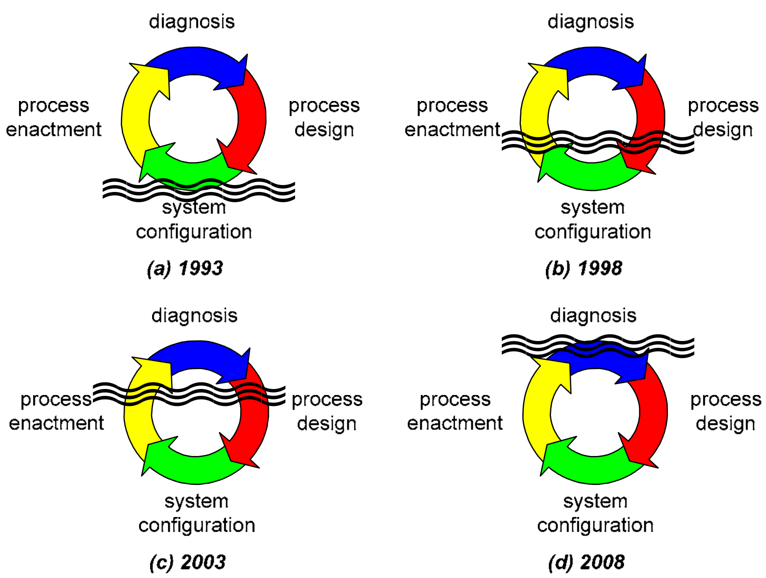

Figure 1: The level of support is rising, i.e., initially only the lower part of the BPM life-cycle was supported. Over time also the design and enactment received more attention. Now the focus is on the diagnosis phase, i.e., using process mining to extract information from logs that can be used for process improvement.

The diagnosis phase assumes that data is collected in the enactment phase. Most information systems provide some kind of *event log* (also referred to as transaction log or audit trail). Typically such an event log registers the start and/or completion of activities. Every event refers to a case (i.e., process instance) and an activity, and, in most systems, also a timestamp, a performer, and some additional data.

Process mining techniques take an event log as a starting point to extract knowledge, e.g., a model of the organization or the process. For example, the ProM (Process Mining) framework (cf. www.processmining.org) developed at Eindhoven University of Technology provides a wide range of process miming techniques. This chapter discusses process mining techniques, and, in particular, the techniques supported by the ProM framework, in the context of decision support.

To link process mining to decision support, we distinguish between four types of decisions when it comes to operational (i.e., workflow-like) processes:

- *Design-time decisions*, i.e., during the initial modeling of a process all kinds of design decisions are made. These decisions are recorded in models and specifications which are used to realize information systems. For example, at design time it may be decided that one activity has to wait for the completion of another because of data dependencies.
- *Configuration-time decisions*, i.e., decisions related to the customization of a process/system for a specific organizational setting. For example, the designers of the SAP R/3 system developed their system based on a set of reference processes describing the different scenarios in which the ERP system can be used. However, to become operational the SAP system needs to be configured for a specific organizational setting. In this configuration process all kinds of decisions are made, e.g., most organizations "switch off" functionality and select the desired mode of operation (e.g., a particular way of invoicing).
- *Control-time decisions*, i.e., decisions to manage processes while they are running. Depending on the context, decisions regarding the use of capacity, the selection of paths, prioritization, etc. are taken. These decisions are at the level of the process and not at the level of an individual process instance but change over time depending on the context. For

example, based on an unusual demand volume in the weeks before Christmas, it is decided not to accept rush orders and capacity from other processes is relocated to the bottlenecks.

- *Run-time decisions,* i.e., decisions made for individual process instances (cases in workflow terminology). These are the decisions typically depicted in process models, e.g., based on the value of an order a particular path through the process is selected. A run-time decision typically depends on the properties of a particular case.

Process mining can assist at all four levels of decision making. At design-time it is very important to know existing processes as they really occur, i.e., an information system design should not be based on an idealized highly idealistic view of processes. Therefore, process mining allows for a reality check (assuming that it is possible to gather event logs). Note that the next release of a system can benefit from past experiences with earlier releases. Moreover, the usability and effectiveness of a system/process can be analyzed using process mining thus supporting redesign decisions. Similarly, process mining can also assist in making configuration decisions. Detailed knowledge of the real use of the system may reveal suboptimal configurations, e.g., there may be functionality that is offered but not used and there may be usage patterns that suggest that people are bypassing the system to get things done. Process mining can also be used to compare different configurations and their effect on performance. Control-time and run-time decisions can also benefit from historic information. For example, process mining tools such as ProM identify bottlenecks in the process and this information can be used to make control-time decisions. Similarly, it is possible to use process mining to make a recommendation service, assisting users in making run-time decisions, e.g., although the process allows for different paths, based on process mining, a subset of paths is recommended because of experience data (e.g., minimize flow times). The examples above illustrate that process mining can be of use at all levels of decision making. The key contribution of process mining is that it provides decision maker a better understanding of the process and models describing reality better. We believe that accurate models are a key requirement for any form of decision support. Therefore, this paper focuses on process mining.

The remainder of the chapter is organized as follows. First, we introduce the concept of process mining and discuss possible applications. Then the ProM framework is used to illustrate the concepts and to zoom in onto the use of process mining for decision support. Finally, we discuss some related work and conclude the paper.

2. Process mining

2.1 Process Mining: An Example

The goal of process mining is to extract information about processes from event logs (also known as audit trails, translation logs, etc.) [3,5]. We assume that it is possible to record events such that (i) each event refers to an *activity* (i.e., a well-defined step in the process), (ii) each event refers to a *case* (i.e., a process instance), (iii) each event can have a *performer* also referred to as *originator* (the person executing or initiating the activity), and (iv) events have a *timestamp* and are totally ordered. In addition events may have associated data (e.g., the outcome of a decision). Events are recorded in a so-called *event log*. To get some idea of the content of an event log consider the fictive log shown in Table 1.

case id	*activity id*	*originator*	*timestamp*
case 1	activity A	John	9-3-2004:15.01
case 2	activity A	John	9-3-2004:15.12
case 3	activity A	Sue	9-3-2004:16.03
case 3	activity D	Carol	9-3-2004:16.07
case 1	activity B	Mike	9-3-2004:18.25
case 1	activity H	John	10-3-2004:9.23
case 2	activity C	Mike	10-3-2004:10.34
case 4	activity A	Sue	10-3-2004:10.35
case 2	activity H	John	10-3-2004:12.34
case 3	activity E	Pete	10-3-2004:12.50
case 3	activity F	Carol	11-3-2004:10.12
case 4	activity D	Pete	11-3-2004:10.14
case 3	activity G	Sue	11-3-2004:10.44
case 3	activity H	Pete	11-3-2004:11.03
case 4	activity F	Sue	11-3-2004:11.18
case 4	activity E	Clare	11-3-2004:12.22
case 4	activity G	Mike	11-3-2004:14.34
case 4	activity H	Clare	11-3-2004:14.38

Table 1: An example of an event log

As we will show later, logs having a structure similar to the one shown in Table 1 are recorded by a wide variety of systems. This information can be used to extract knowledge. For example, the Alpha algorithm [5] described later in this chapter can be used to derive the process model shown in Figure 2.

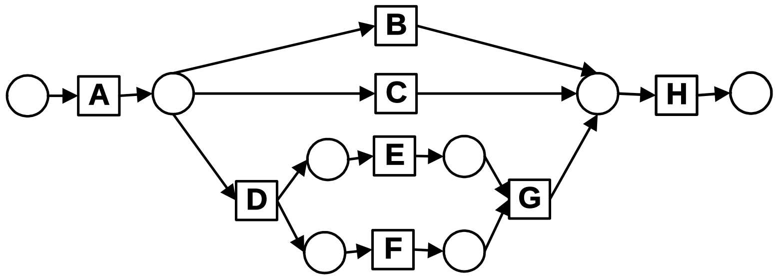

Figure 2: A process model derived from Table 1 and represented in terms of a Petri net.

It is important to note that the Alpha algorithm is just one of the many process mining techniques available. For example, it is possible to extract a social network based on an event log. For some more examples we refer to Section 3.

2.2 Overview of Process Mining and Related Topics

Figure 3 provides an overview of process mining and the various relations between entities such as the information system, operational process, event logs and (process) models.

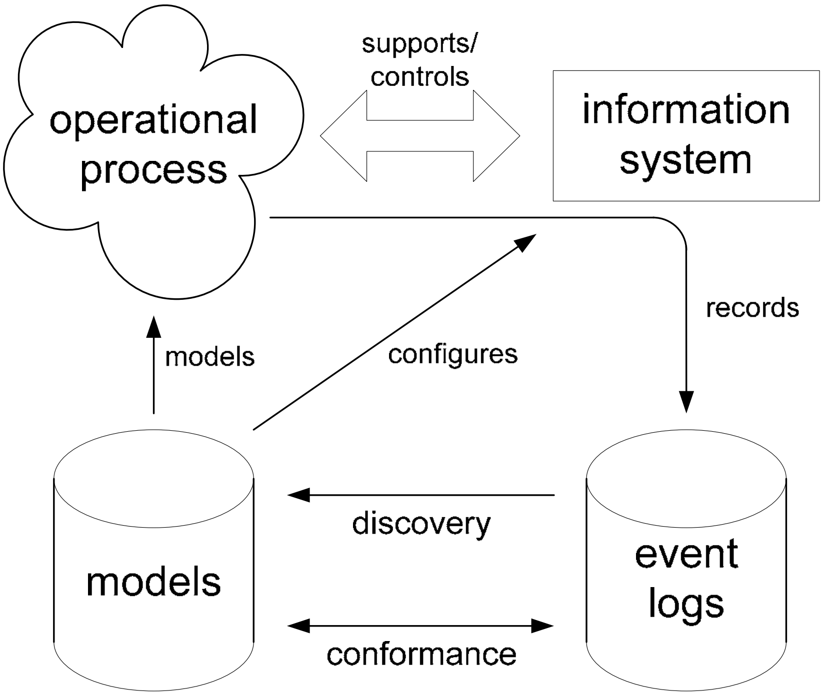

Figure 3: Overview of process mining and related topics.

Figure 3 shows the *operational process* (e.g., the flow of patients in a hospital, the handling of insurance claims, the procurement process of a multinational, etc.) that is interacting with some *information system* (e.g., and ERP, CRM, PDM, BPM, or WFM system). Clearly the information system and the operational process exchange information, e.g., the system may support and/or control the process at hand. The relation between the information system and the operational process is obvious. In the remainder we focus on the role of the *models* and *event logs* shown in Figure 3. After describing their role, we focus on the two arrows related to process mining: *discovery* and *conformance*. As Figure 3 discovery and conformance are in essence concerned with linking models and event logs in the context of an information system and the operational process it supports.

As discussed before, many systems log events related to some process (cf. the arrow labeled *records* in Figure 3). The role of models is more involved. Clearly, process models can be used to model the operational process for a variety of reasons. Process models can be used to analyze and optimize processes but can also be used for guidelines, training, discussions, etc. (cf. the arrow labeled *models* in Figure 3). However, increasingly information systems are configured on the basis of models (cf. the arrow labeled *configures* in Figure 3). For example, consider process-aware systems [8] ranging from production workflow systems such as Staffware and COSA [1] to ERP systems like SAP R/3 and BaaN. Models can be *prescriptive* or *descriptive*. Prescriptive models are somehow used to influence or control the processes while descriptive models are used more for understanding and analyzing the processes. If models are used for configuration, they tend to be prescriptive. If they are used for other purposes, they are often descriptive.

Both the models and the event logs can be seen as some abstraction from the operational process. While event logs record the actual events being logged, the process model focuses at the aggregated level, also referred of as "type level". At this level the goal is not to inspect a single process instance but the collection of all possible/observed instances. The goal of process mining is to extract models from event logs (cf. the arrow labeled *discovery* in Figure 3). Based on the observations recorded in the log, some model is derived. Like in classical data mining it is possible to derive relationships, e.g., causality relations, interaction patterns, and dependencies. Pure process mining just focusing on discovery is complemented by *conformance* checking. Conformance checking is concerned with comparing a model and an event log. This can be used to investigate the fitness and appropriateness of a model (cf. the arrow labeled *conformance* in Figure 3). For example, it can be used to measure alignment. Consider the SAP R/3 reference model expressed in terms of Event-driven Process Chains (EPCs). The EPCs describe best practices, but the SAP system does not enforce people to follow these best practices. Using conformance checking, the actual logs can be compared with the EPCs and indicate where organizations/people deviate. Instead of directly comparing the logs and the models, it is also possible to first do process mining and compare the result with the original model using delta analysis.

2.3 Three Mining Perspectives

Process mining is not restricted to the process perspective (also referred to as control-flow) and also includes other perspectives such as the organizational and data perspectives. In this section, we briefly discuss the three dominant mining perspectives in more detail.

The *process perspective* is concerned with the control-flow, i.e., the causal ordering of activities. In a process model causal relationships are specified, e.g., activity A is followed by activity B, activity C and activity D are in parallel, or after executing activity E there is a choice between activity F and activity G. Consider again Table 1. For the process perspective only the first two columns are relevant and the goal is to derive a process model, e.g., the Petri net shown in Figure 2. To do this we can first translate the table in an audit trail for each case, i.e., case 1: <A,B,H>, case 2: <A,C,H>, 3: <A,D,E,F,G,H>, and case 4: <A,D,F,E,G,H>. Given these traces we apply Occam's Razor, i.e., "one should not increase, beyond what is necessary, the number of entities required to explain anything". This tells us that the process holds activities A, B, C, D, E, F, G, and H. Every process starts with A and end with H. In-between there is a choice between executing (1) B only, (2) C only, or (3) D, E, F, and G. In the latter case, first D is executed followed by both interleavings of E and F, followed by G. Using Occam's principle we deduce that E and F are in parallel. Using a variety of algorithms (e.g., the Alpha algorithm developed by the author [5]) we can deduce the Petri net shown in Figure 2. It is important to note that process mining *should* not require *all* possible observations to be present in the log. This happens to be the case for Table 1/Figure 2, but in general only fraction of the possible behavior will actually be observed. Consider for example a process with 10 binary choices between two alternative activities. In this case one would need to see 2^{10}=1024 different traces. If 10 activities are in parallel, one would need even 10!=3628800 different traces. In such cases one should not expect to see all possible traces, but simply look for the most likely candidate model. This is the reason we are not only using algorithmic approaches and also use heuristics and genetic mining.

The *organizational perspective* is concerned with the organizational structure and the people within the organizational units involved in the process. The focus of mining this perspective is on discovering organizational structures and social networks. Note that Figure 2 completely ignores the third column in Table 1. Nevertheless this column may be used to derive interesting knowledge. For example, it is possible to discover which people typically work together, which people execute similar activities, etc. This can be used to build social networks, i.e., directed graphs where each node represents a person and weighted arcs connecting these nodes represent some relationship.

The *data perspective* is concerned with case and the data associated to cases. Table 1 does not hold any data. However, in reality case and activities have associated data (e.g., the amount of money involved, the age of a customer, the number of order-lines, etc.). Such information may be combined with the columns shown in Table 1 to answer interesting questions such as: "Do large orders take more time than small orders?", "What is the average flow time of cases where John is involved?", "Does the treatment of male patients differ from the treatment of female patients?".

2.4 Obtaining logs

After providing a broad overview of process mining we briefly focus on the nature of the logs that can be obtained in reality. In this chapter, we simply assume a format and then show that many real-life systems have logs that can be converted to this format.

The format that we will use is the so-called MXML (Mining XML) format. The data shown in Table 1 illustrates the nature of this format. However, it is possible to store additional information, e.g., data linked to events, multiple processes at the same time, transactional information, etc.

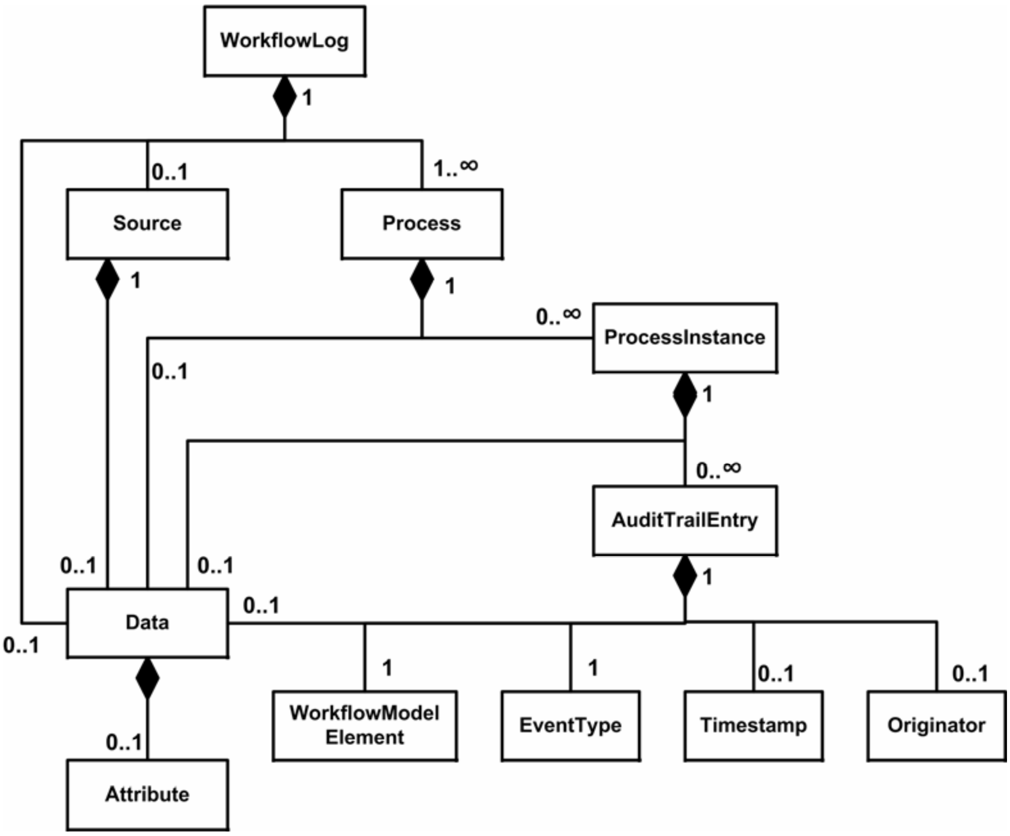

Figure 4: Meta model describing the MXML format.

Figure 4 shows a meta model describing the MXML format. The *Source* element contains the information about software or system that was used to record the log. The *Process* element represents one process holding multiple cases. The *ProcessInstance* elements correspond to cases. One *ProcessInstance* element may hold multiple *AuditTrailEntry* elements. Each of these elements represents an event, i.e., one line in a table like Table 1. Each *AuditTrailEntry* element may contain *WorkflowModelElement*, *EventType*, *Timestamp*, and *Originator* elements. The *WorkflowModelElement* and *EventType* are mandatory elements as shown in Figure 4. The *WorkflowModelElement* element refers to an activity, a subprocess, or some other routing element in the process model. The *EventType* element can be used to record the type of event (e.g., the start or completion of an activity or some exceptional behavior like the cancellation of a case). Table 1 does not show any event types. However, one can always use the default event type *complete*. The *Timestamp* element can be used to record the time of occurrence. The *Originator* element refers to the performer, e.g., the person executing the corresponding activity. To make the format more expressive, we define *Data* element that can be used at various levels (i.e., *WorkflowLog*, *Process*, *ProcessInstance*, and *AuditTrailEntry* level). If users want to specify additional information, this can be recorded using the *Data* element (e.g., data elements linked to cases).

MXML is used by several process mining tools including the ProM framework described in the next section. To create MXML files the ProM Import Framework (cf. http://promimport.sourceforge.net.) has been developed.. The ProM Import Framework has allowed developers to quickly implement import plug-ins for a variety of commercial systems holding suitable logs. Some examples of systems provide logs that ProM or ProM Import can work with are [10]:

- **FLOWer**: This product is an implementation of the *case handling* paradigm, which represents a very flexible, data-driven approach within the greater family of workflow management systems.
- **WebSphere Process Choreographer**: As a part of IBM's WebSphere suite, the Process Choreographer is used to implement high-level business processes, based on the BPEL language.
- **Staffware**: A workflow management system in the traditional sense, which has a big share of the workflow market.
- **PeopleSoft Financials**: Part of the PeopleSoft suite for Enterprise Resource Planning (ERP), this module is concerned with financial administration within an organization.

- **ARIS PPM**: ProM can read three different formats related to EPCs including the instance EPCs provided by the ARIS Process Performance Monitoring (ARIS PPM) tool. This way ProM can access the logs of all tools that are supported by ARIS PPM, e.g., SAP R/3 and many dedicated systems. It is also possible to load models and to import the logs of the simulation tool of ARIS.
- **CPN Tools**: CPN Tools provides excellent tool support for modeling Colored Petri Nets (CPN), a family of high-level Petri Nets, including a simulation engine for executing models. An extension to CPN tools has been developed, allowing to create synthetic event logs during a model simulation.
- **CVS**: The process of distributed software development, as reflected in the commits to a source code repository like CVS, can also be analyzed with techniques from the process mining family.
- **Subversion**: The Subversion system addresses fundamental flaws present in CVS, providing change logs that can also be interpreted by means of process mining.
- **Apache 2**: As the access logs of web servers, like Apache 2, reveal the identity of users from their IP, the exact time and items requested, it is straightforward to distill process event logs from them.

Besides this list of standard systems the ProM Import Framework has been used to convert many company specific logs to MXML. Some examples of ad-hoc event logs that have been generated include [10]:

- The event logs describing the process of patient treatments from raw database tables provided by a large Dutch hospital.
- Production unit test logs from an international manufacturer of IC chip production equipment.
- Conversion of various spreadsheets, e.g., spreadsheets containing patient treatment processes, from an ambulant care unit in Israel and a large Dutch hospital.
- The logs of a Dutch municipality.
- The logs of several Dutch governmental organizations using their own logging formats.

In this chapter we emphasize the mapping of events in various formats to MXML. The reason to do so, is to illustrate that in many application domains suitable event logs are available. Most information systems have databases or logs that can be converted to MXML. The only real requirement is that events need to be linked to process instances. However, in many organizations this is rather easy. For example, almost all events in a hospital are linked to a patient. Therefore, even for the unstructured processes in healthcare it is rather easy to identify process instances using patient id's.

2.5 Using Process Mining for Decision Support

As indicated in the introduction, process mining can be used to get a better understanding of reality and, in our view, this is essential for any form of decision support. Discovery can be used to discover models and conformance can be used to check the validity of models used for decision support. In this paper, we do not propose specific decision support techniques. However, we would like to emphasize that process mining can be used at all four levels of decision making: mentioned in the introduction. Some examples are given below.

- *Design-time decisions*. Using conformance checking and bottleneck analysis one can find out what kind of problems exist in the current system/process. This can be used as input for redesign decisions which may be supported using simulation. For example, ProM is able to automatically generate a simulation model that can be used for evaluating different redesigns.
- *Configuration-time decision.* Process mining can be used to discover different configurations and their effect. It is also possible to verify whether a given configuration fits with the characteristics of the real process. For example, it can be shown that certain enabled features are actually never used.
- *Control-time decision.* Process mining can be used to compare the current state with similar states in the past and suggest ways of dealing with it. For example, it the monitored flow times exceed a certain threshold the process switches to another model.
- *Run-time decisions.* Decisions made for individual process instances can also benefit from past executions. For example, using techniques such as case-based reasoning one can select similar (successful) cases in the past and use these as an example for the handling of new cases. ProM supports a so-called recommendation service which gives advice to a run-time

execution environment (e.g., a workflow management system) to select particular paths based on a detailed analysis of historic information.

To make things more concrete, we describe the ProM system which allows for many forms of process mining. ProM is able to generate all kinds of models and is able to check the conformance of existing models. Hence, it serves as valuable starting point for various forms of decision support.

3. Process Mining using ProM

After developing a wide variety of mining prototypes at Eindhoven University of Technology (e.g., EMiT, Thumb, MinSon, MiMo, etc.), we merged our mining efforts into a single mining framework: the *ProM framework*. Figure 5 shows a glimpse of the architecture of ProM. It supports different systems, file formats, mining algorithms, and analysis techniques. It is possible to add new (mining) plug-ins without changing the framework.

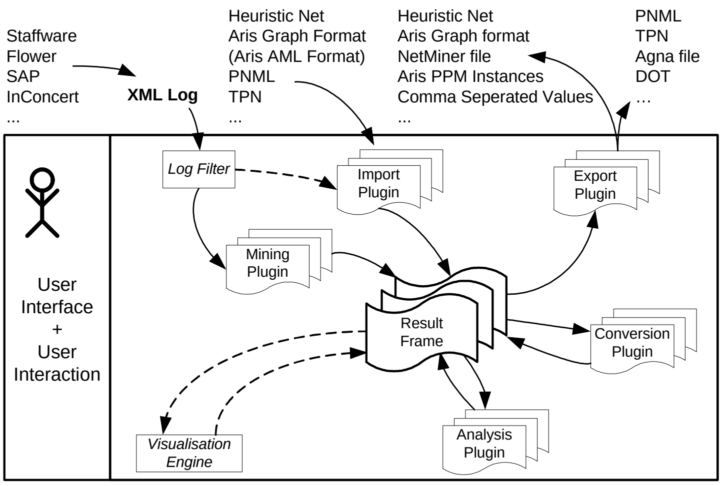

Figure 5: Architecture of ProM.

Currently more than 140 plug-ins are available in ProM. These plug-ins have been realized to offer a wide variety of process mining capabilities. Instead of elaborating on these plug-ins we show some results based on the log shown in Table 1.

Figure 6 shows the result of applying the Alpha algorithm to the event log shown in Table 1. Note that indeed the process shown in Figure 2 is discovered. Since ProM is multi-format it is also possible to represent processes in terms of an EPC or any other format added to the framework.

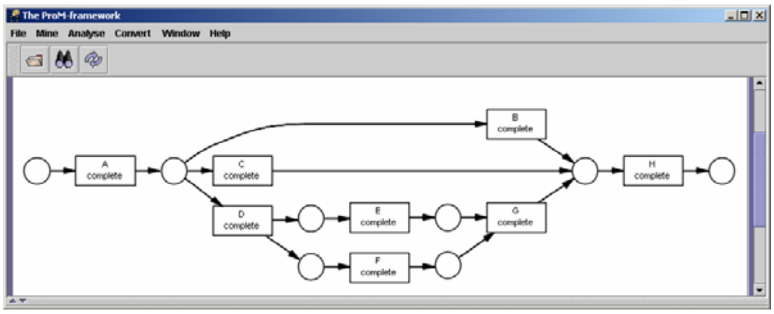

Figure 6: Applying the Alpha plug-in to Table 1

Figure 7 shows a social network based on the event log shown in Table 1. Now nodes represent actors rather than activities [2].

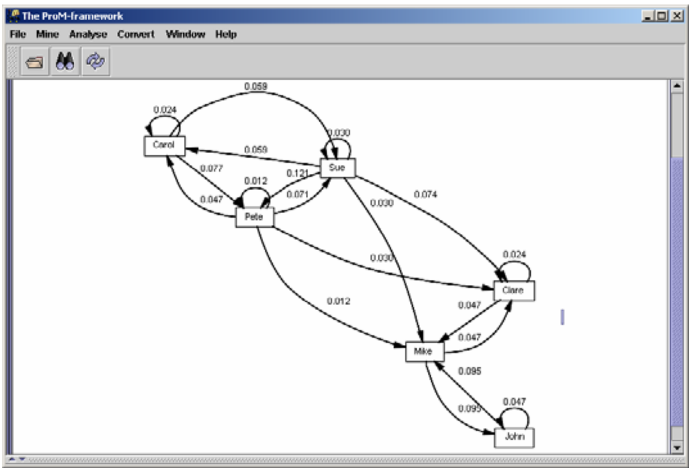

Figure 7: Applying the social network miner plug-in to Table 1.

Figures 6 and 7 show two mining plug-ins that can be used for *discovery*, i.e., the Alpha plug-in aims at discovering the process in terms of a Petri net and the social network miner plug-in aims at discovering the social network. There are many more plug-ins able to discover models describing the process (control-flow), organizational, and data perspectives. To illustrate this we show some more examples using another log that also contains data and time information. The log contains information about the reviewing of papers for a journal. For each paper three reviewers are invited. Reviewers are supposed to return the reviews with a predefined period of time. However, some reviewers do not return the reviews in time. In this case a time-out occurs and the process continues anyway. The reviews are collected and a decision is made. Based on this decision a paper is accepted or rejected. Figure 8 shows a fragment of the corresponding event log.

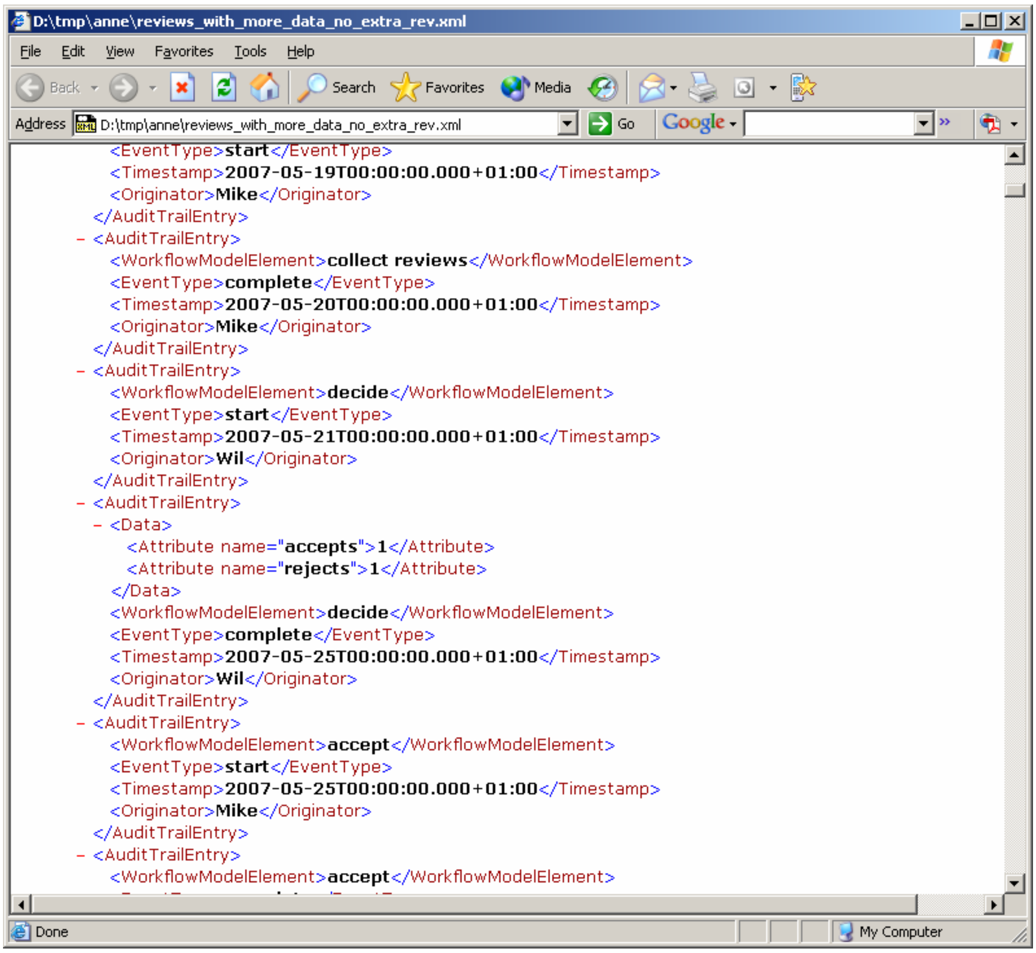

Figure 8: Fragment of MXML log holding information about the reviewing process.

Starting from this event log different process discovery algorithms can be used (e.g., the Alpha algorithm mentioned before [5]). Figure 9 shows the result of applying the multi-phase miner. The resulting model is expressed in terms of an Event-driven Process Chain (EPC). ProM also provides other mining plug-ins, e.g., the heuristics miner and the genetic miner which are able to deal with noise (i.e., logs containing irregular or exceptional behavior) [19].

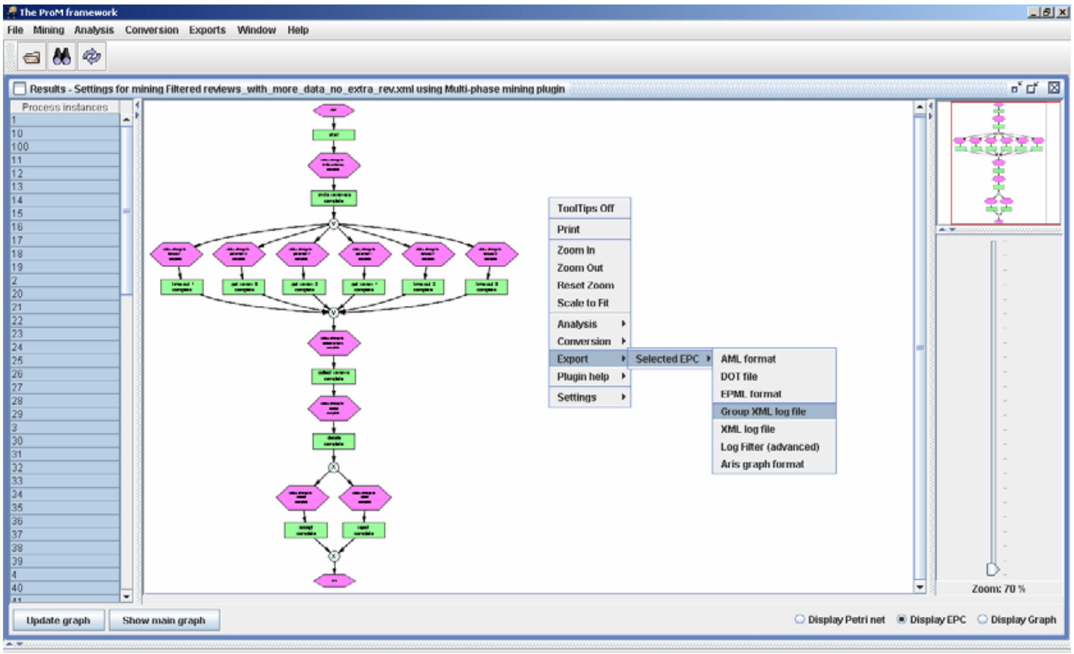

Figure 9: EPC model discovered by the multi-phase plug-in.

As indicated, process mining is not limited to the process (control-flow) perspective. We already showed the social network miner that is able to discover social networks that can be used for organizational analysis. ProM also provides a staff assignment miner that discovers allocation rules based on some organizational model and an MXML log. Figure 10 shows the decision miner when analyzing the choice to accept or reject a paper. This is one of the plug-ins aiming at discovering models for the data perspective [15].

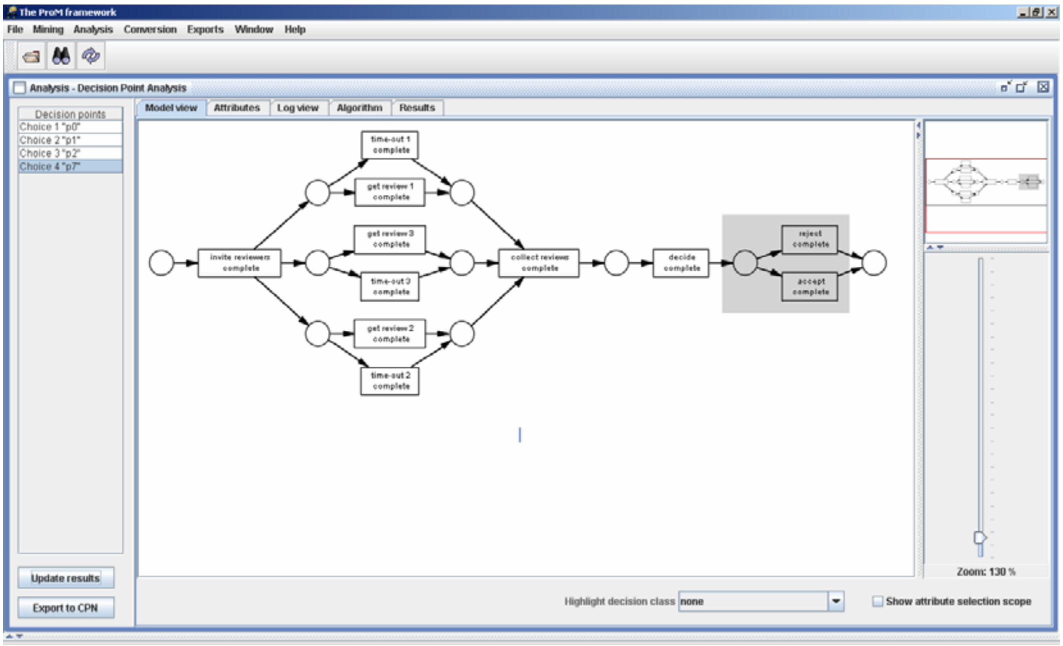

Figure 10: A screenshot of the decision miner while analyzing the choice to accept or reject a paper.

The decision miner takes a discovered process model as a starting point. The Petri net model shown in Figure 10 was discovered using the Alpha algorithm. As can be seen in Figure 8, the log also contains data. This data can be used to analyze in which cases papers follow the path via accept or the path via reject. The decision miner builds a decision tree for this. As shown in Figure 11 papers with more than one reject (i.e., a reviewer voting to reject the paper) are always rejected. If a paper has no or just one rejection, it will be accepted if at least one reviewer votes to accept the paper. Otherwise it is rejected.

Decision mining [15] is highly relevant for decision support since is reveals why certain process instances take a particular path. Moreover, decision mining can also be related to performance information. For example, it may be used to discover that papers that take a long time to review are typically rejected.

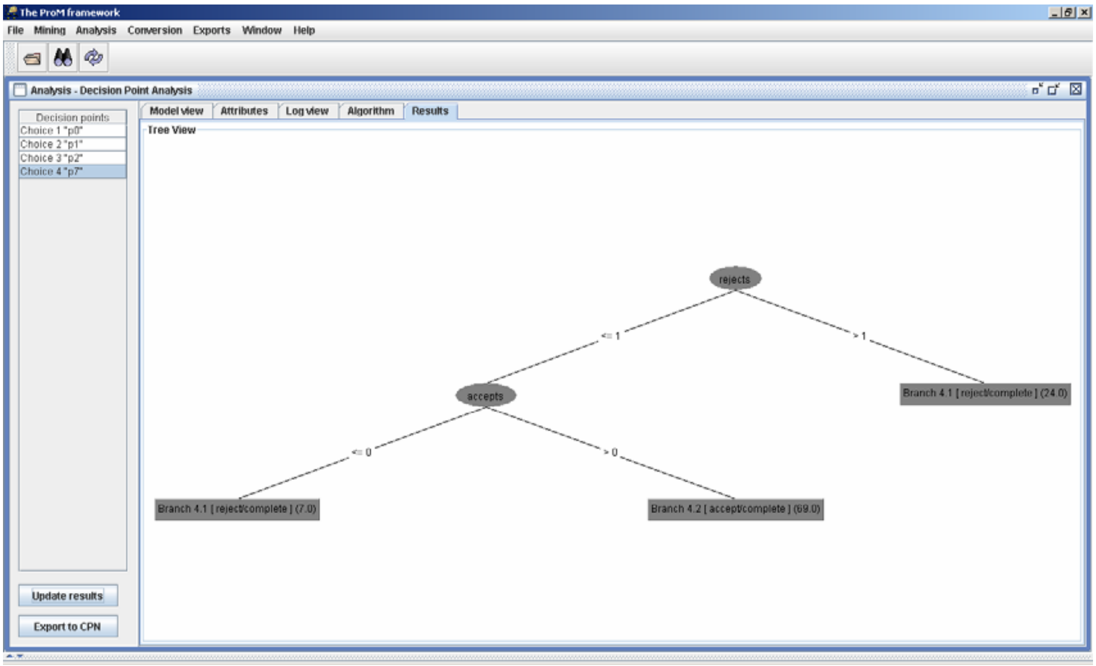

Figure 11: The decision tree describing the choice related to place p7.

ProM also provides plug-ins for performance analysis. Figure 12 shows a plug-in that can visualize the bottlenecks in a process. Performance indicators such as waiting times, service times, flow times, synchronization times, etc. can be derived. *It is important to see that no a-priori modeling is needed to obtain the results depicted in Figure 12.* Existing tools for performance analysis (e.g., ARIS PPM, Business Objects, Cognos, etc.) require the user to define the process before. This means that the user already needs to now the process and the potential bottlenecks.

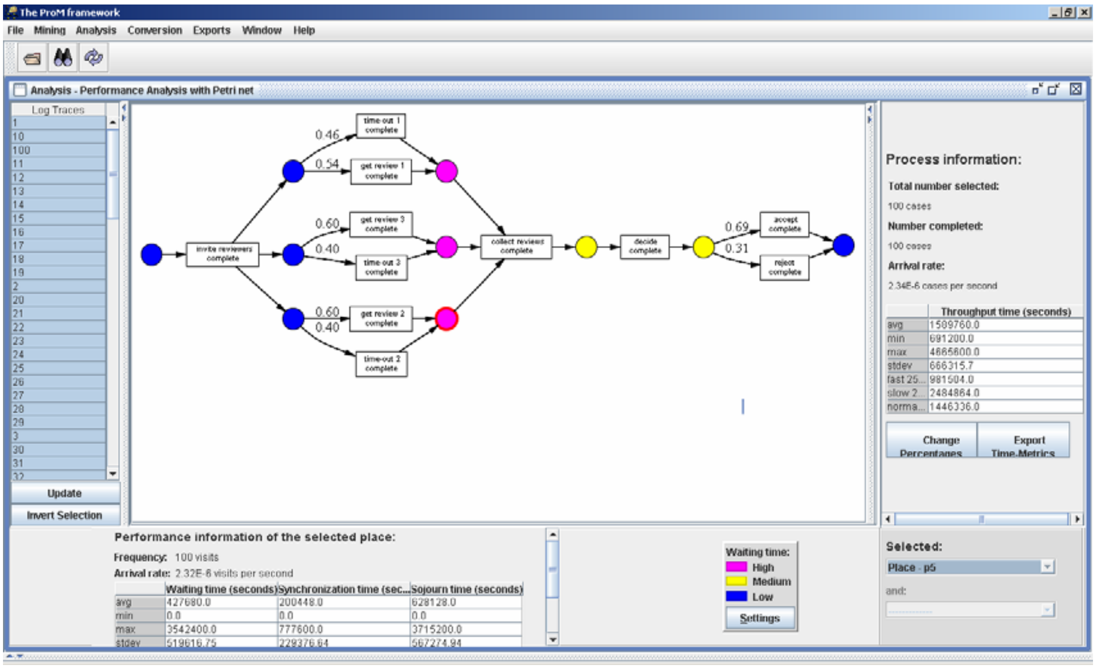

Figure 12: Performance analysis using ProM.

Figures 11 and 12 nicely illustrate how process mining can be used as starting point for decision support. A variety of models can be discovered on the basis of real process executions, i.e., the resulting models are much more objective and reliable than the models typically created by process analysts.

ProM can also be used for conformance checking, i.e., given some a-priori model it is possible to check whether reality is consistent with the model. The a-priori model can be a process model (e.g., an EPC or Petri net) or some business rule. ProM offers a conformance checker [16] that highlights deviations graphically in some a-priori process model. Moreover, any business rule expressed in LTL (Linear Temporal Logic) can be analyzed. For example, it is possible to check which cases follow some four-eyes principle (two activities not to be executed by the same person) or do not meet a given service level agreement (any request is followed by a reply within two weeks).

Also interesting to note is that any process model discovered by ProM can be exported to CPN Tools and YAWL. CPN Tools is a simulation tool that can be used to explore alternative scenarios. YAWL is an open source workflow management system. Moreover, ProM offers a range of process analysis plug-ins (e.g., soundness verification).

For more information on the ProM framework or to download the toolset we refer to www.processmining.org.

4. Related work

The idea of process mining is not new [3,4,6] but has been mainly aiming at the control-flow perspective. The idea of applying process mining in the context of workflow management was first introduced in [6]. This work is based on workflow graphs, which are inspired by workflow products such as IBM MQSeries Workflow (formerly known as Flowmark). Cook and Wolf have investigated similar issues in the context of software engineering processes. In [7] they describe three methods for process discovery: one using neural networks, one using a purely algorithmic approach, and one Markovian approach. Schimm [18] has developed a mining tool suitable for discovering hierarchically structured workflow processes. Herbst and Karagiannis also address the issue of process mining in the context of workflow management using an inductive approach [11,12]. They use stochastic task graphs as an intermediate representation and generate a workflow model described in the ADONIS modeling language. Most of the approaches have problems dealing with parallelism and noise. Our work in [5] is characterized by the focus on workflow processes with concurrent behavior (rather than adding ad-hoc mechanisms to capture parallelism). In [19] a heuristic approach using rather simple metrics is used to construct so-called "dependency-frequency tables" and "dependency-frequency graphs". These are then used to tackle the problem of noise. Process mining is not limited to the control-flow perspective. As shown in [2], it can also be used to discover the underlying social network. In [15] the concept of decision mining is introduced, while in [16] the topic of conformance checking is introduced.

Process mining in a broader sense can be seen as a tool in the context of Business (Process) Intelligence (BPI). In [9,17] a BPI toolset on top of HP's Process Manager is described. The BPI toolset includes a so-called "BPI Process Mining Engine". However, this engine does not provide any techniques as discussed before. Instead it uses generic mining tools such as SAS Enterprise Miner for the generation of decision trees relating attributes of cases to information about execution paths (e.g., duration). In order to do workflow mining it is convenient to have a so-called "process data warehouse" to store audit trails. Such a data warehouse simplifies and speeds up the queries needed to derive causal relations. In [14] Zur Mühlen describes the PISA tool which can be used to extract performance metrics from workflow logs. Similar diagnostics are provided by the ARIS Process Performance Manager (PPM) [13]. The later tool is commercially available and a customized version of PPM is the Staffware Process Monitor (SPM) (www.staffware.com) which is tailored towards mining Staffware logs. Note that none of the latter tools is extracting models, i.e., the results do not include control-flow, organizational or social network related diagnostics. The focus is exclusively on performance metrics. For more information on process mining we refer to a special issue of Computers in Industry on process mining [4] and the survey paper [3].

5. Conclusion

This chapter discussed the application of process mining in the context of decision support. We have show that based on event logs present in wide variety of application domains, we can discover models or check the conformance of existing models. These models may refer to different perspectives. We have shown techniques able to discover process models in terms of Petri nets, EPCs, etc. Based on the same event logs also the data perspective and the organizational perspective can be analyzed. In our view, this is crucial for decision support. Only with accurate models one can truly support decision making. As shown process mining can be used to discover bottlenecks. All of this can be used as a starting point for more traditional decision support approaches, e.g., using optimization and simulation tools.

Although process mining techniques are maturing rapidly and tools such as ProM can easily be applied, there are many open problems and challenges. For example, most of the existing mining techniques have problems dealing with noise and incompleteness. As discussed in this chapter we need to apply Occam's Razor to get meaningful results. (Occam's razor is a logical principle attributed to the mediaeval philosopher William of Occam. The principle states that "one should not increase, beyond what is necessary, the number of entities required to explain anything".) One exception should not change the process model completely and should be ignored or marked as such. Moreover, information will always be based on a limited observation period where not all possible combinations of events will occur. Therefore, it does not make sense to assume a "complete" log.

Besides the "discovery aspect" of process mining, complementary approaches such as delta analysis and conformance testing can be utilized. In particular, conformance testing allows for widespread application. In many settings, it is useful to compare some prescriptive or descriptive model with the actual events being logged.

We hope that this chapter will inspire researchers and developers to apply process mining in new domains. We also encourage people to use the ProM framework as a platform for such efforts. There are interesting links to many forms of decision support. For example, ProM can be linked to workflow management systems to assist in the selection of work-items. Such a recommendation serve has been implemented to offer more support without sacrificing flexibility.

Acknowledgements

The author would like to thank Ton Weijters, Boudewijn van Dongen, Ana Karla Alves de Medeiros, Anne Rozinat, Christian Günter, Minseok Song, Laura Maruster, Eric Verbeek, Monique Jansen-Vullers, Hajo Reijers, Michael Rosemann, Huub de Beer, Ronny Mans, Peter van den Brand, Andriy Nikolov, Wouter Kunst, et al. for their on-going work on process mining techniques. We also thank EIT, NWO and STW for supporting the development of the ProM framework, cf. www.processmining.org.

References

[1] W.M.P. van der Aalst and K.M. van Hee. Workflow Management: Models, Methods, and Systems. MIT press, Cambridge, MA, 2002.

[2] W.M.P. van der Aalst and M. Song. Mining Social Networks: Uncovering Interaction Patterns in Business Processes. In J. Desel, B. Pernici, and M. Weske, editors, International Conference on Business Process Management (BPM 2004), volume 3080 of Lecture Notes in Computer Science, pages 244-260. Springer-Verlag, Berlin, 2004.

[3] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering, 47(2):237-267, 2003.

[4] W.M.P. van der Aalst and A.J.M.M. Weijters, editors. Process Mining, Special Issue of Computers in Industry, Volume 53, Number 3. Elsevier Science Publishers, Amsterdam, 2004.

[5] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128-1142, 2004.

[6] R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In Sixth International Conference on Extending Database Technology, pages 469-483, 1998.

[7] J.E. Cook and A.L. Wolf. Discovering Models of Software Processes from Event-Based Data. ACM Transactions on Software Engineering and Methodology, 7(3):215-249, 1998.

[8] M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley & Sons, 2005.

[9] D. Grigori, F. Casati, M. Castellanos, U. Dayal, and M.C. Shan. Business Process Intelligence, Computers in Industry Journal, Special issue on workflow mining, 53(3): 321-343, 2004.

[10] C.W. Günther and W.M.P. van der Aalst. A Generic Import Framework For Process Event Logs. BPM Center Report BPM-06-13, BPMcenter.org, 2006.

[11] J. Herbst. A Machine Learning Approach to Workflow Management. In Proceedings 11th European Conference on Machine Learning, volume 1810 of Lecture Notes in Computer Science, pages 183-194. Springer-Verlag, Berlin, 2000.

[12] J. Herbst. Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen. PhD thesis, Universität Ulm, November 2001.

[13] IDS Scheer. ARIS Process Performance Manager (ARIS PPM): Measure, Analyze and Optimize Your Business Process Performance (whitepaper). IDS Scheer, Saarbruecken, Gemany, http://www.ids-scheer.com, 2002.

[14] M. zur Mühlen and M. Rosemann. Workflow-based Process Monitoring and Controlling - Technical and Organizational Issues. In R. Sprague, editor, Proceedings of the 33rd Hawaii International Conference on System Science (HICSS-33), pages 1-10. IEEE Computer Society Press, Los Alamitos, California, 2000.

[15] A. Rozinat and W.M.P. van der Aalst. Decision Mining in Business Processes. BETA Working Paper Series, WP 164, Eindhoven University of Technology, Eindhoven, 2006.

[16] A. Rozinat and W.M.P. van der Aalst. Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. In C. Bussler et al., editor, BPM 2005 Workshops (Workshop on Business Process Intelligence), volume 3812 of Lecture Notes in Computer Science, pages 163-176. Springer-Verlag, Berlin, 2006.

[17] M. Sayal, F. Casati, U. Dayal, and M.C. Shan. Business Process Cockpit. In Proceedings of 28th International Conference on Very Large Data Bases (VLDB'02), pages 880-883. Morgan Kaufmann, 2002.

[18] G. Schimm. Generic Linear Business Process Modeling. In S.W. Liddle, H.C. Mayr, and B. Thalheim, editors, Proceedings of the ER 2000 Workshop on Conceptual Approaches for E-Business and The World Wide Web and Conceptual Modeling, volume 1921 of Lecture Notes in Computer Science, pages 31-39. Springer-Verlag, Berlin, 2000.

[19] A.J.M.M. Weijters and W.M.P. van der Aalst. Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integrated Computer-Aided Engineering, 10(2):151-162, 2003.

Index Terms

- process mining
- Petri nets
- social networks
- workflow management
- Petri nets
- conformance
- discovery

Profile

Prof.dr.ir. Wil van der Aalst is a full professor of Information Systems at the Technische Universiteit Eindhoven (TU/e) having a position in both the Department of Mathematics and Computer Science and the department of Technology Management. Currently he is also an adjunct professor at Queensland University of Technology (QUT) working within the BPM group. His research interests include workflow management, process mining, Petri nets, business process management, process modeling, and process analysis.
For more information on ongoing projects where Wil van der Aalst in involved in, see:

www.processmining.org
www.workflowpatterns.com
www.workflowcourse.com
www.yawl-system.com
www.bpmcenter.org

Contact information:

Prof.dr.ir. W.M.P. van der Aalst
Eindhoven University of Technology
Department of Mathematics and Computer Science
PO Box 513
NL-5600 MB Eindhoven

The Netherlands

Phone:
+31 40 247.4295

Fax:
+31 40 243.2612

E-mail:
w.m.p.v.d.aalst@tue.nl

WWW:
http://is.tm.tue.nl/staff/wvdaalst/