



Process Mining—Discovery, Conformance, and Enhancement of Manufacturing Processes

Stefanie Rinderle-Ma, Florian Stertz, Juergen Mangler and Florian Pauker

Abstract

Process-orientation has gained significant momentum in manufacturing as enabler for the integration of machines, sensors, systems, and human workers across all levels of the automation pyramid. With process orientation comes the opportunity to collect manufacturing data in a contextualized and integrated way in the form of process event logs (no data silos) and with that data, in turn, the opportunity to exploit the full range of process mining techniques. Process mining techniques serve three tasks, i.e., (i) the discovery of process models based on process event logs, (ii) checking the conformance between a process model and process event logs, and (iii) enhancing process models. Recent studies show that particularly, (ii) and (iii) have become increasingly important. Conformance checking during run-time can help to detect deviations and errors in manufacturing processes and related data (e.g., sensor data) when they actually happen. This facilitates an instant reaction to these deviations and errors, e.g., by adapting the processes accordingly (process enhancement), and can be taken as input for predicting deviations and errors

S. Rinderle-Ma (✉) · J. Mangler
Department of Informatics, TU München, Munich, Germany
e-mail: stefanie.rinderle-ma@tum.de

J. Mangler
e-mail: juergen.mangler@tum.de

F. Stertz
Faculty of Computer Science, Universität Wien, Vienna, Austria
e-mail: florian.stertz@univie.ac.at

F. Pauker
EVVA Sicherheitstechnologie GmbH, Vienna, Austria
e-mail: f.pauker@evva.com

for future process executions. This chapter discusses process mining in the context of manufacturing processes along the phases of an analysis project, i.e., preparation and analysis of manufacturing data during design and run-time and the visualization and interpretation of process mining results. In particular, this chapter features recommendations on how to employ which process mining technique for different analysis goals in manufacturing.

Keywords

Manufacturing Processes • Process Discovery • Conformance Checking • Process Enhancement

1 Introduction

“Recent trends in automation and knowledge of the underlying processes / interactions are key to digital transformation” [14]. In manufacturing, process technology has already proven itself as a driver for digital transformation. *Manufacturing processes*—aka manufacturing orchestrations—integrate the manufacturing tasks conducted by human actors, sensors, machines, and information systems in a process-oriented way [19]. This integration, in turn, enables the contextualized collection of process-related data and hence facilitates getting full transparency on what is going on using *process mining* [24].

Figure 1a1) depicts an example manufacturing process realizing the production of a piece for a gas turbine (i.e. lowerhousing) (cmp. [9]). The manufacturing process is modeled using Business Process Modeling and Notation (BPMN).¹

The BPMN model starts off with the sub process Turn1 represented by a complex task, followed by another turning and two milling sub processes. Afterwards a quality control (QC) task takes place at the shopfloor. If the quality is not OK the process is completed, otherwise a QC task at the customer side takes place. This decision is reflected by an alternative branching.

The BPMN model reflects the control flow of the manufacturing process, i.e., it abstracts from aspects such as data flow, resources, and time. The BPMN model can then be imported into a process execution engine such as the Cloud Process Execution Engine (CPEE)² where it is transformed into an executable model by specifying, for example, endpoints and data (the CPEE model is shown in Fig. 1a3). For the given manufacturing process, for example, each manufacturing process instance reflects one work piece to be produced.

Information on the execution of the manufacturing process instances is collected during run-time by the process engine and stored in *process event logs* (cf. Fig. 1a4). Note that also other information systems such as ERP systems collect (process) event logs and process event logs reflect *event-based data*. The latter means that for each manufacturing process

¹ bpmn.org

² cpee.org

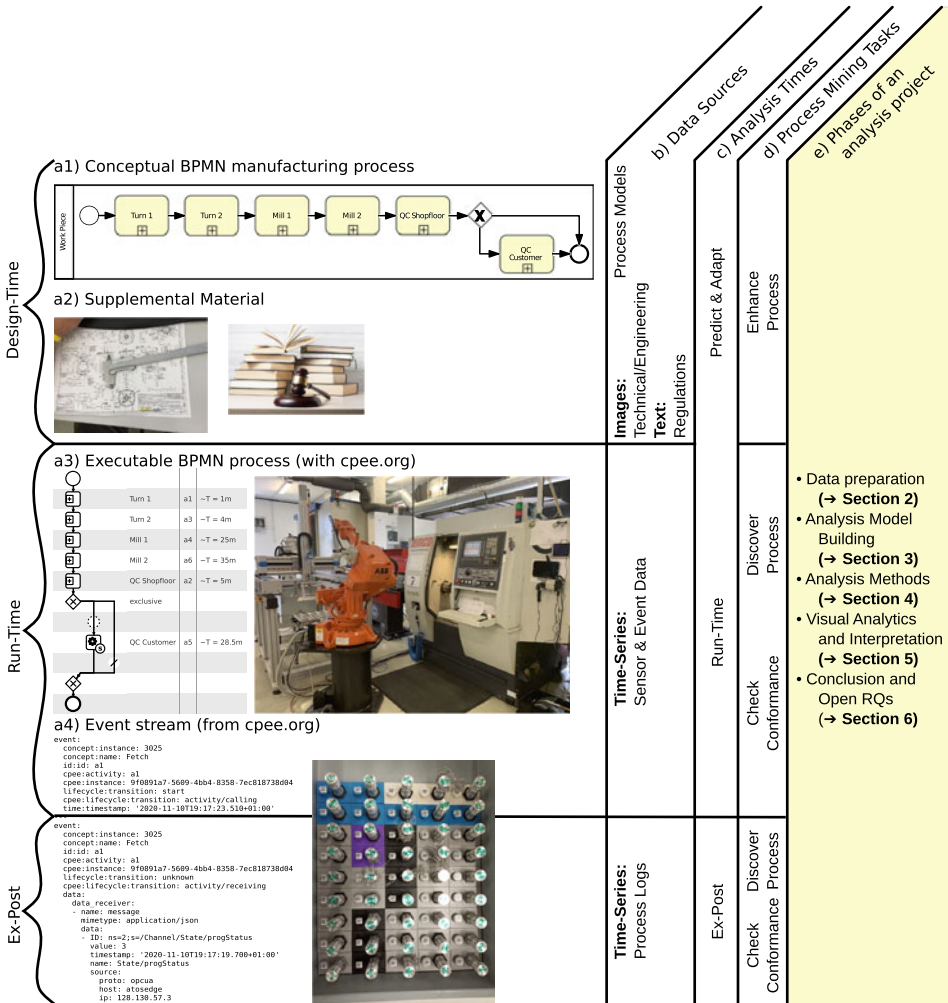


Fig. 1 Manufacturing Scenario & Structure of Chapter

task (e.g., QC Shopfloor in Fig. 1a1 and a3) at least one event reflecting its execution is stored, together with a time stamp, and information on the process instance the process task was executed for. This is the minimum information required for conducting process mining. Several extensions might be stored and analyzed such as events on start/completion of tasks, data and resources. In general, the more and better³ data is available, the better the analysis results might turn out. The standard format for process event log is eXtensible

³ “Better” here refers to the quality of the collected data. For a discussion on quality levels of process event logs see Sect. 2.

Event Stream (XES) [1]. In addition to process event logs, the following data sources might provide insights to the manufacturing process (cf. Fig. 1b): (i) sensor and machine data that is collected during the process execution, stored as *time series*; (ii) engineering drawings, stored as images, and (iii) regulations such as ISO standards, stored as text. A deeper insight into the different data sources and how to prepare them will be provided in Sect. 2.

Process event logs are the basic input for the application of *process mining*. Process mining is particularly promising for manufacturing processes as “*transparency is a prerequisite for digital transformation*” and “*process mining allows full transparency based on event logs*” as stated in [24]. Consequently, process mining seems highly promising for gaining a deeper understanding of manufacturing processes and for promoting the digitalization in the manufacturing domain. The three tasks of process mining are *process discovery*, *conformance checking*, and *process enhancement* [2, 6] (cf. Fig. 1d). In short, process discovery refers to detecting process models from process event logs, conformance checking to the assessment of conformance/deviations between process models and event logs, and process enhancement to the adaptation and improvement of process models based on process mining results.

Typically, process mining is conducted *ex post*, i.e., after process instances have been completed and the associated process event logs have been collected (cf. Fig. 1c). Recently, process mining during the *run-time* of process instances has gained momentum as it enables to react to analysis results, e.g., deviations, more quickly [35]. Note that for run-time process mining, literature also refers to *process event streams* instead of process event logs in order to emphasize the run-time/online character of the analysis and hence to *streaming process mining* [5]. There is a fluent transition from run-time/streaming process mining to *predictions*. The latter has been addressed by the area of *predictive process monitoring* [36, 37]. These approaches focus on predicting (i) the next activity to be executed and (ii) the remaining execution time of processes. In manufacturing, *predictive maintenance*, see for example [22], has been a hot topic since several years.

Due to the integration of processes, sensors, machines, human workers, and information systems, manufacturing processes can collect data from all these sources in an *integrated and contextualized way*, i.e., process event logs/streams as event-based data, sensor and machine data as time series data, technical/engineering drawings as images, and regulations as text as depicted in Fig. 1a2 and b). In the light of this abundance of data sources and the opportunities that come with them, in the following we will examine the question of how to analyze manufacturing processes through process mining along the phases of a data analysis project as depicted in Fig. 1e): data preparation in Sect. 2, analysis model building in Sect. 3, analysis techniques in Sect. 4, as well as visualization and interpretation in Sect. 5. Finally, Sect. 6 provides a short summary of the state of the art and discusses future questions.

2 Data Preparation

The input data for process mining are *process event logs* [2]. They consist of a set of *process traces* that reflect sequences of events related to the execution of process activities. More precisely, for each process instance that was created, instantiated, and executed, a process trace reflects the event sequence produced by executing the activities of this process instance.

This section addresses the preparation of process events logs as input for process mining in the manufacturing domain. In Sect. 2.1, we comment on data quality. Section 2.2 explains which data sources are available and how they can be exploited for process mining.

2.1 Data Quality in Manufacturing

As stated for business intelligence projects in general [10] and process mining projects in particular [7], the collection and preparation of data is often the most complex and tedious task. This also holds true for the manufacturing domain where—without explicit process orientation and data collection—the data accrues over several systems and system layers along the automation pyramid as depicted in Fig. 2 (middle). The right side of Fig. 2 assigns the systems at the different layers of the automation pyramid to the quality classes of the L* data quality model as established in the Process Mining Manifesto [7]. The L* model features five quality levels from * (lowest) to ***** (highest) based on criteria such as event orientation, trustworthiness, and systematic collection.

Enterprise resource planning (ERP) data (***) is event-oriented and can be assumed as trustworthy. ERP data is collected in a systematic way, but provides only a specific view on the data, e.g., on the production orders. The data of lower layers of the automation pyramid is assessed as ** data meaning that event data is collected “*as a by-product of some information system*” [7]. As a consequence the information system might be bypassed resulting in missing and/or incorrect data. The lower the layer, the more unclear is the quality level (at most **). Moreover, the data collection mostly happens in a separated manner, i.e., the data is not collected across the layer in a contextualized ways. This leads to data silos. If, by contrast, the data is collected in a process-oriented way as shown on the left side of Fig. 2, the data is at least of quality level ****, i.e., event-oriented, collected in a systematic way, and trustworthy. If the data is in addition semantically annotated, it can be classified as of highest quality (*****).

Figure 3a depicts a process event log provided as eXtensible Event Stream (XES) (xes-standard.org). The log contains a trace for process instance 423. It is crucial that traces have unique ids such that the information on the activity executions can be distinguished for different instances, e.g., activity Turn1 was executed for instance 423 and not for instance 424. Further on, the trace carries information on the process engine the process instance was executed with, i.e., the CPEE², together with a UUID. The trace contains two

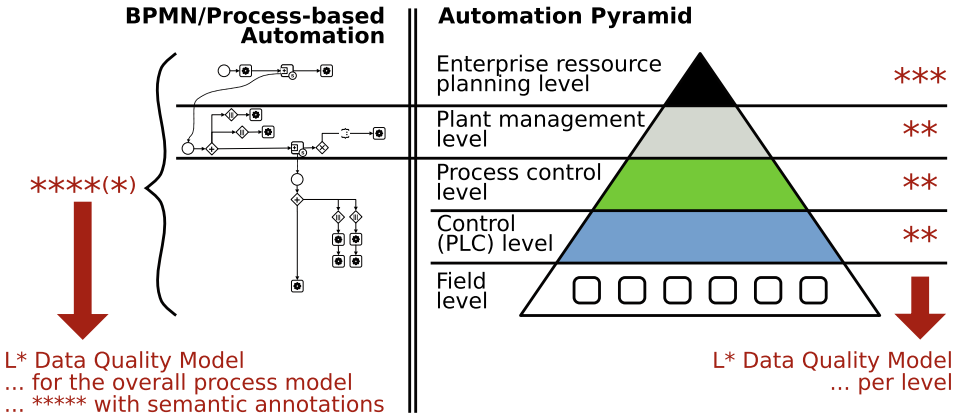


Fig. 2 Data collection along automation pyramid and in a process-oriented manner with data quality level according to L^* model [7]

a) Process Event Log (XML-based XES):

```
<log xes.features="" xes.version="2.0">
...
<trace>
<string key="concept:name" value="1794"/>
<string key="cpee:name" value="GV12 Turn Production"/>
<string key="cpee:uuid" value="2c8ec53c-9315-42e3-ad02-b5ca3a771b22"/>

<event>
<string key="concept:instance" value="1794"/>
<string key="cpee:uuid" value="2c8ec53c-9315-42e3-ad02-b5ca3a771b22"/>
<string key="concept:name" value="Machining"/>
<string key="cpee:endpoint" value="https://cpee/flow/start/url"/>
<string key="id:id" value="a1"/>
<string key="lifecycle:transition" value="start"/>
<string key="cpee:lifecycle:transition" value="activity/calling"/>
<date key="time:timestamp" value="2018-10-18T10:05:10.722+02:00"/>
</event>
<event>
<string key="concept:instance" value="1794"/>
<string key="cpee:uuid" value="2c8ec53c-9315-42e3-ad02-b5ca3a771b22"/>
<string key="concept:name" value="Machining"/>
<string key="cpee:endpoint" value="https://cpee/flow/start/url"/>
<string key="id:id" value="a2"/>
<string key="lifecycle:transition" value="complete"/>
<string key="cpee:lifecycle:transition" value="activity/done"/>
<date key="time:timestamp" value="2018-10-18T10:08:58.408+02:00"/>
</event>
</trace>
</log>
```

b) Process Event Log (YAML-based XES):

```
log:
...
trace:
  concept:name: 1794
  cpee:name: GV12 Turn Production
  cpee:uuid: ecdf5d27-08a9-4830-8faa-4faf8383e1e6

event:
- concept:instance: 1794
  cpee:uuid: ecdf5d27-08a9-4830-8faa-4faf8383e1e6
  concept:name: Machining
  concept:endpoint: https://cpee/flow/start/url/
  id:id: a1
  lifecycle:transition: start
  cpee:lifecycle:transition: activity/calling
  time:timestamp: '2019-11-14T19:31:21.461+01:00'

- concept:instance: 1794
  cpee:uuid: ecdf5d27-08a9-4830-8faa-4faf8383e1e6
  concept:name: Machining
  concept:endpoint: https://cpee/flow/start/url/
  id:id: a2
  lifecycle:transition: complete
  cpee:lifecycle:transition: activity/done
  time:timestamp: '2019-11-14T19:34:38.742+01:00'
```

Fig. 3 Manufacturing process event logs in **a** XES-XML and **b** XES-YAML formats

events that refer to the execution of process activity with label Machining. Note that for more complex scenarios, traces may contain several thousands or more events.

The two events referring to the execution of Machining distinguish between the start and the completion of Machining, i.e., refer to two activity life cycle states [34]. Often, process event logs only store one event per activity execution, mostly for their com-

pletion. However, if life cycle states can be distinguished in the logs, more precise analysis results can be achieved, for example, on the duration of activities. In addition to `start` and `completion` of process activities life cycle states such as `interrupted` exist. This can be also seen in the more fine-granule CPEE life cycle states such as `activity/calling`. Finally, the events are equipped with a `timestamp` which is vital in order to reason about the activity orders, e.g., in a discovered process model.

However, as said before, more information can be stored, leading to more options during the analysis. The start event of activity `Machining` in Fig. 3a, for example, holds additional information on its `end_point`, i.e., the service that is called for executing the activity. Further information typically stored in process event logs are `resource/originator`, i.e., the actor or machine that has performed the task.

There are several ways to represent a serialization of the XES format. XML is usually used, since it is human readable, provides a schema and can be easily processed. The example in Fig. 3a features an XML representation of an process event log. Figure 3b presents another approach to serialize the XES format in YAML. YAML is as human readable as XML [27] and offers advantages over XML for directly logging events in the XES format since the computational effort is lower because new events, instead of parsing an XML tree for the correct position, can be easily appended to a file, an operation which in many operating systems is optimized.

2.2 Data Sources and Process Mining

Table 1 summarizes existing process mining approaches along the analysis time they are applied at, i.e., Ex post, Run-Time, and Predict&Adapt, (cf. Fig. 1c), as well as along the available data sources, i.e., process event logs containing different amount of information. For the latter, we start from the minimum necessary information to be able to apply process mining, and step-by-step add information (indicated by a + in Table 1).

The different analysis methods and techniques shown in Table 1 will be explained in Sect. 4. In this section, we focus on the data preparation. The rows above the one highlighted in light blue in Table 1 refer to data sources that can be entirely captured within a process event logs. On top of the minimum necessary information, in the log we distinguish between start and end of an event or other life cycle transitions. With start and end events it becomes possible to reason about the duration of activities (e.g., [31]). Moreover, process event logs can also contain information on resources, see the organizational extension in XES.⁴ This enables to analyze organizational structures connected with the process, for example, underlying work or social networks [28] as well as authorization rules (who is allowed to perform which process activity) [18]. If the process event log also stores values of process data such as temperature that are associated with process activities, these values can be exploited for finding decision rules at alternative branchings (decision discovery [16]). One

⁴ xes.standard.org

Table 1 Data input and existing approaches

Trace contains \ Analysis phase	Ex post	Runtime	Predict&Adapt
Minimum information: trace id, event name, 1 life cycle state, time stamp	Control flow discovery, conformance checking		Next activity, remaining time
+ start/end event	Activity durations, temporal profiles [31]		
+ resource/originator	Organizational mining [28], mining of actor assignments [18]		
+ (internal) process data	Decision discovery [16]	Data drift detection [33], multi perspective conformance checking [20]	
+ (external) sensor and machining data	Decision discovery [8]	Explaining and predicting concept drift [35]	
+ multimedia data (text, image, video)	Decision discovery [16]		

example would be: if the temperature is below 30°C the machine works in normal mode, otherwise a cooling agent has to be used. We refer to this process data and the associated values as *internal data*. During run-time, process data can also be exploited for detecting drifts in the process data [33] and for multi-perspective conformance checking [20].

It is crucial to be able to establish links between process activities and the sensor/machining data. This leads to the question how to deal with sensor and machining data. This data can be used as internal data, i.e., be necessary for executing the process, but—because this is continuous or time series data—it cannot be directly stored in the process event log. This leads to the question on how to store such “internal/external” data, so it can be used for process mining. Sensors are creating data points continuously even without an active process, i.e., like the temperature in the previous example. There are at least three options available to use this data in process mining (cf. [8, 35]). The first option aims at adapting the process mining techniques to take a log containing time series data points as an additional input, so that with the temporal information of the process event log, the related data points to an event can be extracted. The other option would be to adapt the process model, to add a task, which fetches data points from a sensor. As a third option, to achieve results during run-time, this fetching task can also be inserted in a parallel loop to the process, so that the data points are inserted into the log while the process is being executed.

Manufacturing processes can be also subject to multimedia data sources (see last row in Table 1), including texts such as regulatory documents, standards, and norms, (ii) image data such as technical or engineering drawings as input or for quality control, and video data such as instructions. So far, existing approaches have provided means for exploiting data values of primitive data types, i.e., numbers and strings for determining decision rules in an ex post manner [16]. Support for more complex data types or other multimedia data is currently missing (see future work in Sect. 6).

As discussed in the Process Mining Manifesto [7], in practice, process data is often not available in an event-based format. This also holds true for the manufacturing domain where the data is often scattered over several information systems (e.g., ERP) and the machines. Process-orientation offers the opportunity to integrate these “data silos” resulting in an contextualized collection of process event logs [19].

3 Analysis Model Building

According to [10] “[f]inding answers for analytical goals is based on models”. Hence, modeling is an essential task in a manufacturing analysis project. Further on, [10] states that model building can be understood “as a mapping of some part of the domain semantics of the business process into the model structure. This happens in such way that the available data enable formal analysis of questions about the process”. Here, the domain is manufacturing, the processes are the manufacturing processes, and the available data consists of process event logs plus potentially additional data as outlined in Table 1.

For process mining, the central analysis model is a *graph* which represents the control flow of the discovered process models. The graph is typically described using existing process meta models such as BPMN in Fig. 1a1 and a3) as well as Petri Nets and Heuristic Nets in Fig. 4a and b. Note that other analysis models for discovered manufacturing processes are conceivable such as patterns [10] for, e.g., mining declarative process models (for an overview on declarative process mining approaches see [17]). Graph-based structures can be also used as analysis models beyond the control flow of processes, for example, social networks as model for organizational structures underlying a manufacturing process (cf. Fig. 4c). This underpins that depending on the data available in the process event log (cf. Table 1) and the analysis question different analysis models might become viable.

In particular for the manufacturing domain, the analysis of process event logs in combination with additional (external) data such as sensor or machining data is vital [35]. The question is which analysis model can be used as basis for the combined analysis. Sensor and machining data is available in the form of time series data. As discussed in [10], time series data is produced by collecting the states of one or several variables over time and can be analyzed based on statistical models (cf. Fig. 4d).

4 Analysis Methods

The analysis tasks of process mining are process discovery, conformance checking, and process enhancement [2]. These analysis tasks together with existing techniques are discussed in the following in the context of manufacturing processes.

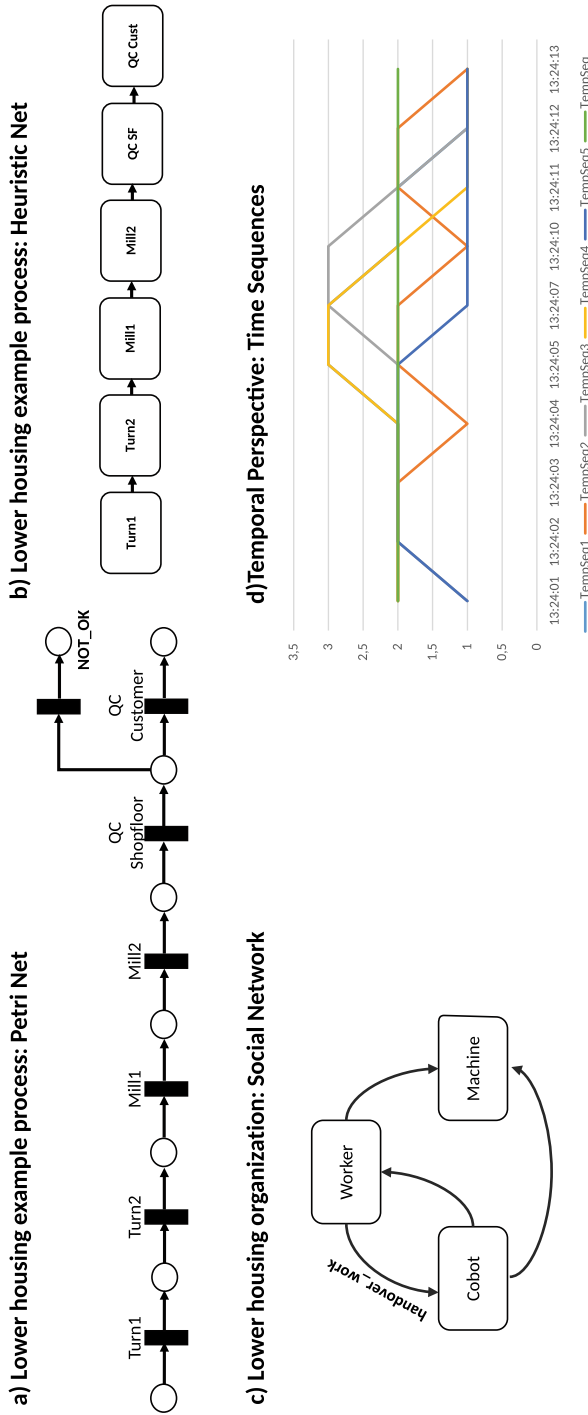


Fig. 4 Analysis models for process mining (selection): **a** Petri Nets and **b** Heuristic Nets and **c** Social Network and **d** Temporal Perspective

Discovery of manufacturing processes: Basically, the control flow of manufacturing processes can be discovered using existing algorithms such as the α miner [21], the Heuristic miner [38], or the Inductive miner [15].

Consider the `Spawn GV12 Production` process execution log⁵ which consists of 81 traces. An excerpt is shown in Fig. 3b. Figure 5 depicts the Heuristic net that is discovered when applying the Heuristic miner (using PM4PY 2.2.4⁶) based on this log. The Heuristic miner considers order relations between pairs of activities a and b that can be observed in the log and calculates their relative frequency, i.e., how often does b directly follow a minus the how often the opposite happens (i.e., a directly follows b) divided by the sum of b follows a and a follows b . This leads to a number between 0 and 1. The higher the number, the more likely it is that b actually follows a . This formula is exemplarily calculated for the relation between `GV12 Turn9 (start)` and `GV12 Turn9 (complete)`. The result suggests the actual order `GV12 Turn9 (start) \Rightarrow GV12 Turn9 (complete)`.

Figure 5 shows interesting results. There are splits in the discovered Heuristic net, i.e., the order of `Manually Measure9 (start)` and `Signal Machining End9 (complete)` is often times mixed up. The same phenomenon can be witnessed with `Manually Measure9 (complete)` and `Measure with MicroVu9 (start)`. This is a clear indication to dig deeper into these cases, i.e., traces, and emphasizes the capabilities of process discovery as screening tool (this observation has been already made in the context of medical treatment processes [3]).

Figure 6 shows the Petri net resulting from applying the Inductive miner infrequent (using PM4PY) on the `Spawn GV12 Production` process event log. The discovered Petri net confirms that with this log, `Manually Measure9 (start)` and `Signal Machining End9 (complete)` appear to be executed in parallel. Based on this visual inspection of both discovered models, we can go back to the traces and dig deeper into the reason for the observation on `Measure with MicroVu`. Using, for example, the log filtering capabilities of PM4PY, we can see that only nanoseconds are separating these events. Therefore a faulty behavior in the log system is likely the origin of these errors.

Conformance of manufacturing processes: Conformance checking [6] takes as input a process model and a process event log and calculates the conformance between the log and the model or—vice versa—the difference between them. More precisely, the goal is to measure to which degree the behavior expressed by the process model (i.e., all producible traces on the model) and the behavior stored in the log (i.e., all traces in the log) match. If we understand the process model as the expression of the intended behavior and the log as collection of the actual behavior, conformance assesses how much reality is reflected by the intended/prescribed behavior in the model. Note that there might be real-world behavior that is neither reflected by the process model nor (already) stored in the log.

In manufacturing, conformance checking can be used for monitoring the process behavior over time [9], i.e., monitoring whether and how actual process execution conforms or deviates

⁵ <http://gruppe.wst.univie.ac.at/data/timesequence.zip>

⁶ <https://pm4py.fit.fraunhofer.de/>

$$\frac{|\text{GV12 Turn9 (start)} \succ_w \text{GV12 Turn9 (complete)}| - |\text{GV12 Turn9 (complete)} \succ_w \text{GV12 Turn9 (start)}|}{|\text{GV12 Turn9 (start)} \succ_w \text{GV12 Turn9 (complete)}| + |\text{GV12 Turn9 (complete)} \succ_w \text{GV12 Turn9 (start)}| + 1} = 75/76 = 0.986$$

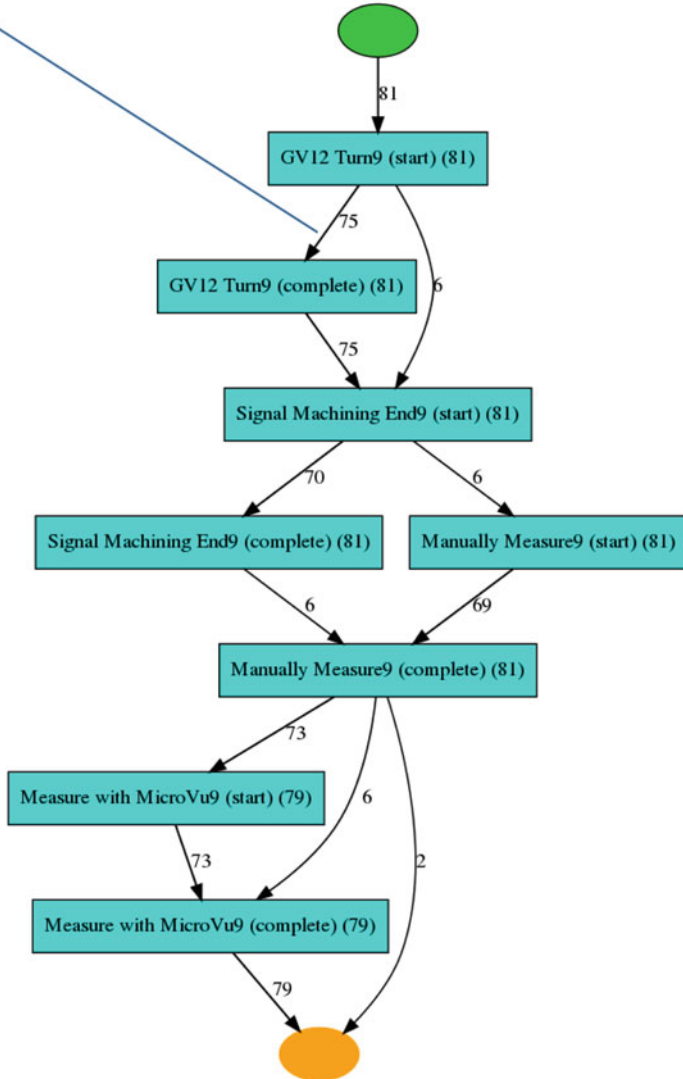


Fig. 5 Spawn GV12 Production log using Heuristic Miner yielding Heuristic net



Fig. 6 Spawn GV12 Production log using Inductive Miner yielding a Petri net with infrequent transitions removed

from the originally specified model. If the manufacturing processes are already implemented and executed through a process engine, deviations from the process model, i.e., at the control flow level, can only occur due to adaptations of either single process instances (ad hoc adaptations) or due to an evolution of the entire process model [23]. Deviations in process behavior that might happen for various reasons are also referred to as *concept drift* [4, 32]. Being able to analyze, explain, and predict concept drifts is of utmost interest in general and specifically in the manufacturing domain [35].

One “component” of quantifying conformance between a process event log and a process model is *fitness* [26]. Fitness measures how much of the behavior stored in the log is reflected by the process model. This value can be calculated by virtually replaying the behavior expressed by the traces of the process event log on the given process model. For this the process model is represented as a Petri net and the replay of the traces is carried out by replaying tokens on the Petri net. Then it is counted how often the replay requires extra tokens for continuing the execution or creates missing ones. Figure 7 shows the `Spawn GV12 Production` example. As input we take the original process execution log containing 81 traces (for an excerpt see Fig. 3b). As discussed in the previous paragraph on process discovery, this log contains traces for which some tasks appear in the wrong order due to an inaccuracy of the logging system. The created Petri Net (cf. Fig. 6) only features frequent transitions and therefore does not guarantee perfect fitness. Hence the log does feature traces, which are not fitting the model. A small excerpt showing the results is shown in Fig. 7b. As can be seen, `batch5_55` yields a fitness less than 1 and therefore is not fit.

Another measure to capture conformance is *precision* which states how much of the behavior expressed by the process model is also reflected in the log, i.e., the model should not allow additional event sequences which are not reflected in the process event log [2]. The difference between the process event log and the process model can be also measured in terms of adaptation operations that would be necessary to transform the log to the model or vice versa. As can be seen from Fig. 7c the Petri net (also called flower model) can produce any trace on the 4 contained transitions reflecting manufacturing tasks.⁷ A selection of example traces is also depicted in Fig. 7c. Obviously, the fitness of, for example, the original log `Spawn GV12 Production` (cf. Fig. 3b) would be 1. However, the log does not reflect all traces that can be generated in the flower model. This fact can be reflected by measuring precision.

Aside other limitations as discussed in [6], fitness and precision only measure “one side of the coin”, i.e., either how much behavior of the log is also reflected by the model or vice versa. In addition, in particular for the manufacturing domain, it would be more interesting to present conformance results in a more expressive way than providing a value between 0 and 1. Take, for example, a fitness value of 0.85. It means that the behavior of the log is reflected by the model in a “pretty high” manner, but it does neither allow any conclusions where deviations between model and log occur nor any assessment how severe these deviations are.

⁷ As a simplification we only included the tasks without differentiation into start/complete tasks.

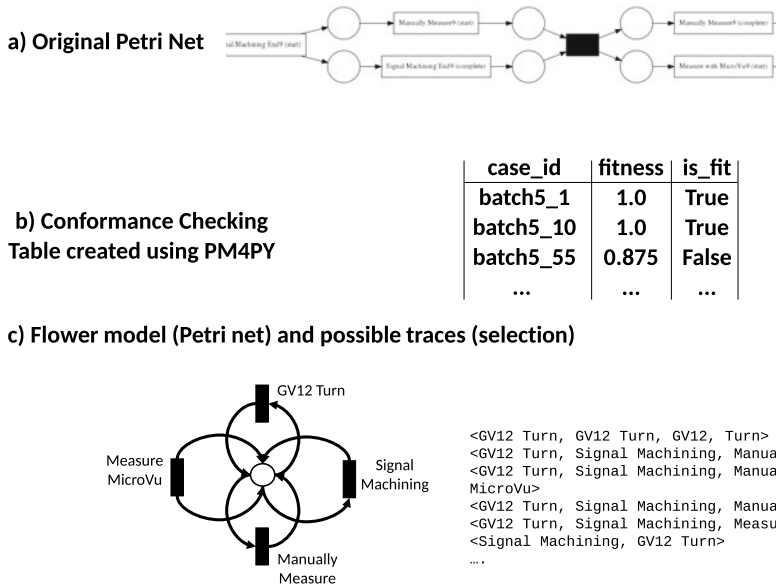


Fig. 7 Spawn GV12 Production: conformance checking on process model without infrequent transitions (transformed to Petri net model using Inductive miner, PM4PY) and original Spawn GV12 Production process event logs (using Conformance Checking, PM4PY)

Alignments [6] promise to overcome these limitations. At first, alignments can be used to capture deviations between log and model. More precisely, an alignment states which adaptation steps/operations are necessary to “correct” a deviation between an execution trace from the log and a possible execution sequence on the model. Taking all alignments together reflects a sequence of adaptations to transform the log and the model such that they perfectly match/conform. Moreover, the alignments can be equipped with costs.

As an example take the following execution trace t from a manufacturing log and the execution sequence s of a manufacturing process model:

$$t = \langle \text{Turn1}, \text{Turn2}, \text{Mill1}, \text{Mill2}, \text{QC} \rangle \text{ and}$$

$$s = \langle \text{Turn1}, \text{Turn2}, \text{Mill1}, \text{Mill2}, \text{MicroVu} \rangle$$

An alignment between t and s is defined as a matrix that captures t and s and an operator \gg . This operator indicates an adaptation that is necessary to correct a deviation. The result for t and s is shown in Table 2. We can see 4 *synchronous moves* for events in t and s referring to the same activity, e.g., Turn1. The operator \gg is present twice in order to balance the presence of task QC in t and task MicroVu in s .

Typically, an alignment is not unique, i.e., it is possible to transform an execution trace and an execution sequence such that they match using different alignments. Obviously, it is desirable to find a minimum alignment, i.e., an alignment with minimum number of moves.

Table 2 Alignment between process execution trace and execution sequence on manufacturing process model (example)

t	Turn1	Turn2	Mill1	Mill2	QC	>
s	Turn1	Turn2	Mill1	Mill2	>>	MicroVu

This is also sufficient as moves can be assigned costs in order to assess an alignment in a quantitative way. In other words, the costs express how “expensive” it is to align the execution trace and sequence. Here, the costs for the minimum alignments are meaningful, i.e., the minimum costs for not being conformant.

A straightforward way is to assign a cost of 1 to each move. The cost of the alignment shown in Table 2 then turns out as 2. More sophisticated cost functions are conceivable, considering, for example, the influence of moves on data elements and vice versa. In [29], we presented an advanced cost function based which costs for moves indicating missing events can be reduced if the data elements still possess valid values. Such a situation might hint to a logging error where a task was actually executed and has hence manipulated the values of associated data elements, but the corresponding event has not been logged. The real-world production case in [29] underpins that control flow deviations, e.g., swapped executions of tasks, can be tolerated in case the values of relevant parameters, i.e., the final positioning of the produced parts on a tray, are in an acceptable range.

As indicated in Table 1, conformance checking can take place at either design time, i.e., ex post based on process event logs, or at run-time, i.e., online based on event streams. Experience in the manufacturing domain shows that online conformance checking is promising in order to support the continuous monitoring of the production, i.e., the manufacturing process instances. Moreover, it can be observed that control flow deviations happen due to ad-hoc adaptations of single process instances (e.g., in order to cope with exceptional situations) or due to evolution of the manufacturing process model, e.g., if errors or problem with the process occur frequently. On top of control flow deviations, shifts in sensor and machining data [35] as well as temporal deviations [31] yield valuable insights into possible deviations during run-time

Enhancement of manufacturing processes refers to (constantly) adapt and optimize the manufacturing process models and their instances. Process enhancement has gained the highest momentum recently according to the Gartner study in [14] when compared to the other process mining tasks discovery and conformance checking. However, relatively little attention has been spent on process enhancement in literature. In prior work [35], we combined results from online conformance checking in sensor data with explaining and predicting *concept drifts* (cf. Table 1) where the predicted concept drifts are realized by process evolutions. In detail, the measurements of the part (reflected by sensor data) showed significant drifts which could be explained by the occurrence of chips as by-product of the machining task. This led to the identification of several process enhancements, for example,

the insertion of a dedicated `Remove Chips` task in the manufacturing process. A recent study [30] in the manufacturing domain found that conformance checking combined with the analysis of sensor data (*data-aware conformance checking*) supports the detection and explanation of problems with the quality of the workpiece whereas conformance checking focusing on time deviations (*time-aware conformance checking*) points to problems with work organization.

In general, deviations can be detected in a relatively easy and systematic manner. The main challenges are to explain/understand the deviations (i.e., to find and communicate the root cause) and to derive suitable enhancement actions. For the latter, the body of work from process change and evolution (cf. [23] for an overview) can be utilized. Approaches that aim at explaining change and supporting users in defining change operations comprise the augmentation of change operations with reasons [25] as well as the mining of change processes [11] and change stress [12] from change logs.

5 Visual Analytics and Interpretation

Visual analytics is used throughout all phases of the manufacturing analysis project (cf. Fig. 1e) including explorative analysis of the (raw) data, understanding and discussing analysis results, and presenting the final results to the domain experts and other stakeholders. According to [13], basically, “*in the Engineering domain, Visual Analytics can contribute to speed-up development time for products, materials, tools and production methods [...] One key goal [...] in the engineering domain will be the analysis of the complexity of the production systems in correlation with the achieved output, for an efficient and effective improvement of the production environments.*”. This general assessment of visual analytics in the engineering domain can be directly transferred to visual process analytics in manufacturing.

According to [10], one can look at the data from different perspectives, i.e., from the customer, organization, and production perspective. These perspectives are connected with different views on the data (i.e., cross-sectional, state, and event view) and subsequently different analysis techniques (i.e., data mining, processes mining, social network analysis, etc.) based on different analysis models such as graphs. Table 3 displays the different perspectives together with the visualization options.

For the customer perspective, different types of data can be distinguished, together with different visualization options. All of these visualization options might be relevant for the manufacturing domain, particularly for visualizing parameter values. Sensor data, for example, can be visualized as time series plot, i.e., displaying the sensor values over time.

For the production perspective that corresponds to possible results of process mining in terms of process models, basically, we distinguish between design time and run-time visualization. Where design time requires the visualization of process model as graphs following some process meta model (e.g., BPMN), during run-time, the information about

Table 3 Overview on visualization options along the different analysis perspectives

Perspective	Visualizations	
Customer	Qualitative Data	Bar Charts, Pie Charts, Mosaic Plots, ...
	Quantitative Data	Contour Plots, Boxplots, ...
	Relationships	Scatter Plots, Correlation, ...
	Temporal Data	Time series Plots, Survival Plots, ...
Organization	Graph visualizations	
Production	Design Time	Visualizations of Petri Nets, BPMN
	Runtime	Equipping process models with runtime information, displaying log information additionally to process model

the execution progress/state of the different instances is depicted, as well. Here, basically, there are two options: process models can be equipped by run-time information such as colored tokens on Petri Nets or the process model is displayed alongside with process event logs.

6 Conclusion and Open Research Questions

The following recommendations for conducting a process mining project in the manufacturing domain conclude this chapter:

- Use process discovery as screening tool
- Use conformance checking for detecting deviations
- Use data-aware conformance checking (i.e., combined with the analysis of sensor data) for
 - detecting and explaining workpiece quality problems
 - explaining and predicting concept drift
 - predictive maintenance
- Use time-aware conformance checking for detecting temporal deviations and problems with work organization
- Explain and discuss detected deviations with domain experts in order to define remedy actions (⇒ process enhancement)

Nonetheless, several research and application-oriented questions are still open and seem promising for future work, not only for manufacturing, but for any process. Table 1 depicts research directions in light pink color. These research directions mainly refer to the analysis phase of Predict&Adapt for input log data beyond control flow, i.e., including information on resources and internal/external data and to the exploitation of external data. First approaches on exploiting sensor data in addition to log data have been presented (e.g., [35]). The observation here is that this also results in a combination of process and data mining/machine learning techniques (e.g., conformance checking in combination with dynamic time warping). The exploitation of data sources such as text or images in combination with process mining has not been explored yet, although these data sources play an important role in manufacturing (i.e., ISO standards and engineering drawings) and other domains (standard operating procedures and images in medicine).

References

1. IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams (Nov 2016)
2. van der Aalst, W.M.P.: *Process Mining - Data Science in Action*, Second Edition. Springer-Verlag Berlin Heidelberg (2016)
3. Binder, M., Dorda, W., Duftschmid, G., Dunkl, R., Fröschl, K.A., Gall, W., Grossmann, W., Harmankaya, K., Hronsky, M., Rinderle-Ma, S., Rinner, C., Weber, S.: On analyzing process compliance in skin cancer treatment: An experience report from the evidence-based medical compliance cluster (EBMC2). In: *Advanced Information Systems Engineering*. pp. 398–413 (2012)
4. Bose, R.J.C., Van Der Aalst, W.M., Zliobaite, I., Pechenizkiy, M.: Dealing with concept drifts in process mining. *IEEE Trans. Neural Netw. Learning Syst.* 25(1), 154–171 (2014)
5. Burattin, A.: Streaming process discovery and conformance checking. In: *Encyclopedia of Big Data Technologies* (2019)
6. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: *Conformance Checking - Relating Processes and Models*. Springer (2018), <https://doi.org/10.1007/978-3-319-99414-7>
7. van Der Aalst, W., et al.: *Process mining manifesto*. In: *Business Process Management*. pp. 169–194. Springer (2011)
8. Dunkl, R., Rinderle-Ma, S., Grossmann, W., Fröschl, K.A.: A method for analyzing time series data in process mining: application and extension of decision point analysis. In: *CAiSE Forum*. pp. 68–84 (2014)
9. Ehrendorfer, M., Fassmann, J., Mangler, J., Rinderle-Ma, S.: Conformance checking and classification of manufacturing log data. In: *Business Informatics*. pp. 569–577 (2019)
10. Grossmann, W., Rinderle-Ma, S.: *Fundamentals of Business intelligence*. Springer-Verlag Berlin Heidelberg (2015)
11. Günther, C.W., Rinderle-Ma, S., Reichert, M., van der Aalst, W.M.P., Recker, J.: Using process mining to learn from process changes in evolutionary systems. *Int. J. Bus. Process. Integr. Manag.* 3(1), 61–78 (2008)
12. Kaes, G., Rinderle-Ma, S.: Mining and querying process change information based on change trees. In: *Service-Oriented Computing*. pp. 269–284 (2015)

13. Keim, D.A., Andrienko, G.L., Fekete, J., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: Definition, process, and challenges. In: *Information Visualization - Human-Centered Issues and Perspectives*, pp. 154–175 (2008)
14. Kerremans, M., Searle, S., Srivastava, T., Iijima, K.: *Market guide for process mining* (2020), www.gartner.com
15. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Process and deviation exploration with inductive visual miner. In: *BPM Demo*. p. 46 (2014)
16. de Leoni, M., Mannhardt, F.: Decision discovery in business processes. In: *Encyclopedia of Big Data Technologies* (2019)
17. Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: Compliance monitoring in business processes: Functionalities, application, and tool-support. *Inf. Syst.* 54, 209–234 (2015)
18. Ly, L.T., Rinderle, S., Dadam, P., Reichert, M.: Mining staff assignment rules from event-based data. In: *Business Process Management Workshops*. pp. 177–190 (2005)
19. Mangler, J., Pauker, F., Rinderle-Ma, S., Ehrendorfer, M.: centurio.work - industry 4.0 integration assessment and evolution. In: *BPM Industry Forum*. pp. 106–117 (2019)
20. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Balanced multi-perspective checking of process conformance. *Computing* 98(4), 407–437 (2016)
21. de Medeiros, A.K.A., van der Aalst, W.M.P., Weijters, A.J.M.M.: Workflow mining: Current status and future directions. In: *On The Move to Meaningful Internet Systems*. pp. 389–406 (2003)
22. Mobley, R.: *An Introduction to Predictive Maintenance*. Elsevier (2002)
23. Reichert, M., Weber, B.: *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*. Springer (2012)
24. Reinkemeyer, L.: *Process Mining in Action – Principles, Use Cases and Outlook*. Springer International Publishing (2020)
25. Rinderle, S., Weber, B., Reichert, M., Wild, W.: Integrating process learning and process evolution - A semantics based approach. In: *Business Process Management* pp. 252–267 (2005)
26. Rozinat, A., Van der Aalst, W.M.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* 33(1), 64–95 (2008)
27. Shadiya, P., Haleem, P.A.: Energy efficient data formatting scheme: A review and analysis on xml alternatives. *Energy* 1(1) (2012)
28. Song, M., van der Aalst, W.M.P.: Towards comprehensive support for organizational mining. *Decis. Support Syst.* 46(1), 300–317 (2008)
29. Stertz, F., Mangler, J., Rinderle-Ma, S.: Data-driven improvement of online conformance checking. In: *Enterprise Distributed Object Computing*. pp. 187–196 (2020)
30. Stertz, F., Mangler, J., Rinderle-Ma, S.: The role of time and data: Process mining in the manufacturing domain. *Business Information Systems Engineering* (2020), (submitted to)
31. Stertz, F., Mangler, J., Rinderle-Ma, S.: Temporal conformance checking at runtime based on time-infused process models. *CoRR abs/2008.07262* (2020)
32. Stertz, F., Rinderle-Ma, S.: Process histories – detecting and representing concept drifts based on event streams. In: *Cooperative Information Systems*. pp. 318–335 (2018)
33. Stertz, F., Rinderle-Ma, S.: Detecting and identifying data drifts in process event streams based on process histories. In: *CAiSE Forum*. pp. 240–252 (2019)
34. Stertz, F., Rinderle-Ma, S., Hildebrandt, T., Mangler, J.: Testing processes with service invocation: Advanced logging in CPEE. In: *Service-Oriented Computing*. pp. 189–193 (2016)
35. Stertz, F., Rinderle-Ma, S., Mangler, J.: Analyzing process concept drifts based on sensor event streams during runtime. In: *Business Process Management*. pp. 202–219 (2020)

36. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Trans. Knowl. Discov. Data* 13(2), 17:1–17:57 (2019)
37. Verenich, I., Dumas, M., Rosa, M.L., Maggi, F.M., Teinemaa, I.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Trans. Intell. Syst. Technol.* 10(4), 34:1–34:34 (2019)
38. Weijters, A., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP 166, 1–34 (2006)