Process Mining. Data science in action

Julia Rudnitckaia

Brno, University of Technology, Faculty of Information Technology,
irudnickaia@fit.vutbr.cz

Abstract. At last decades people have to accumulate more and more data in different areas. Nowadays a lot of organizations are able to solve the problem with capacities of storage devices and therefore storing “Big data”. However they also often face to problem getting important knowledge from large amount of information. In a constant competition entrepreneurs are forced to act quickly to keep afloat. Using modern mathematics methods and algorithms helps to quickly answer questions that can lead to increasing efficiency, improving productivity and quality of provided services. There are many tools for processing of information in short time and moreover all leads to that also inexperienced users will be able to apply such software and interpret results in correct form.

One of enough simple and very powerful approaches is Process Mining. It not only allows organizations to fully benefit from the information stored in their systems, but it can also be used to check the conformance of processes, detect bottlenecks, and predict execution problems.

This paper provides only the main insights of Process Mining. Also it explains the key analysis techniques in process mining that can be used to automatically learn process models from raw event data. Generally most of information here is based on Massive Open Online Course: "Process Mining: Data science in Action".

Keywords: Big Data, Data Science, Process Mining, Operational processes, Workflow, BPM, Process Models

1. Introduction

Data science is the profession of the future, because organizations that are unable to use (big) data in a smart way will not survive. Clive Humby even offered to use metaphor “Data is new oil” to emphasize what important role the data plays nowadays [7]. It is not sufficient to focus on data storage and data analysis. The data scientist also needs to relate data to process analysis. Process mining bridges the gap between traditional model-based process analysis (e.g., simulation and other business process management techniques) and data-centric analysis techniques such as machine learning and data mining.

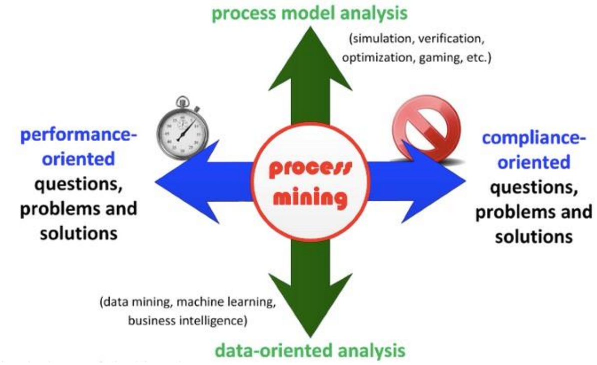

process model analysis
(simulation, verification, optimization, gaming, etc.)
performance-oriented questions, problems and solutions
process mining
compliance-oriented questions, problems and solutions
(data mining, machine learning, business intelligence)
data-oriented analysis

Figure 1. Positioning of PM

More and more information about business processes is recorded by information systems in the form of so-called “event logs”, that is start point for process mining. Although event data are omnipresent, organizations lack a good understanding of their actual processes. Management decisions tend to be based on PowerPoint diagrams, local politics, or management dashboards rather than careful analysis of event data. The knowledge hidden in event logs cannot be turned into actionable information. Advances in data mining made it possible to find valuable patterns in large datasets and to support complex decisions based on such data. However, classical data mining problems such as classification, clustering, regression, association rule learning, and sequence/episode mining are not process-centric.

Process mining seeks the confrontation between event data (i.e., observed behavior) and process models (hand-made or discovered automatically). This technology has become available only recently, but it can be applied to any type of operational processes (organizations and systems). Example applications include: analyzing treatment processes in hospitals, improving customer service processes in a multinational, understanding the browsing behavior of customers using a booking site, analyzing failures of a baggage handling system, and improving the user interface of an X-ray machine. All of these applications have in common that dynamic behavior needs to be related to process models.

2. Overview of Process Mining

There is an overview [1, 2, 3] that concludes main interests of PM:

2.1. Process Mining. Basic concept

Process mining (PM) techniques are able to extract knowledge from event logs commonly available in today's information systems. These techniques provide new means to discover, monitor, and improve processes in a variety of application domains. There are two main drivers for the growing interest in process mining:

1) more and more events are being recorded, thus, providing detailed information about the history of processes;
2) there is a need to improve and support business processes in competitive and rapidly changing environments.

Process Mining provides an important bridge between BI and BPM, Data Mining and Workflow. It includes (automated) process discovery (i.e., extracting process models from an event log), conformance checking (i.e., monitoring deviations by comparing model and log), social network/ organizational mining, automated construction of simulation models, model extension, model repair, case prediction, and history-based recommendations.

In the table below there are pointed various directions of analysis and significant questions that PM can answer to. [6]

№	Use Case	Questions	Group of questions
1	Detection of real-world business processes	What is the process that actually (and not in words and not in theory) describes current activities?	coherence
2	Search bottlenecks in business processes	Where are places in the process, limiting the overall speed of its implementation? What causes these places?	productivity
3	Detection of deviations in business processes	Where the actual process deviates from the expected (ideal) process? Why are there such deviations?	coherence
4	Search fast / short cuts execution of business processes	How to perform the process of the fastest? How to perform a process for the least amount of steps?	productivity
5	Prediction of problems in business processes	Is it possible to predict the occurrence of delays / deviations / risks / ... in the performance of the process?	Productivity/ coherence

Table 1. PM use cases

2.2. Types of PM

There are three main types of process mining (Fig. 2, 3).

1. The first type of process mining is **discovery**. A discovery technique takes an event log and produces a process model without using any a-priori information. An example is the Alpha-algorithm that takes an event log and produces a process model (a Petri net) explaining the behavior recorded in the log.
2. The second type of process mining is **conformance**. Here, an existing process model is compared with an event log of the same process. Conformance checking can be used to check if reality, as recorded in the log, conforms to the model and vice versa.
3. The third type of process mining is **enhancement**. The main idea is to extend or improve an existing process model using information about the actual process recorded in some event log. Whereas conformance checking measures the alignment between model and reality, this third type of process mining aims at changing or extending the a-priori model. An example is the extension of a process model with performance information, e.g., showing bottlenecks.

In the Figure 2 it's shown PM types in terms of inputs and outputs.

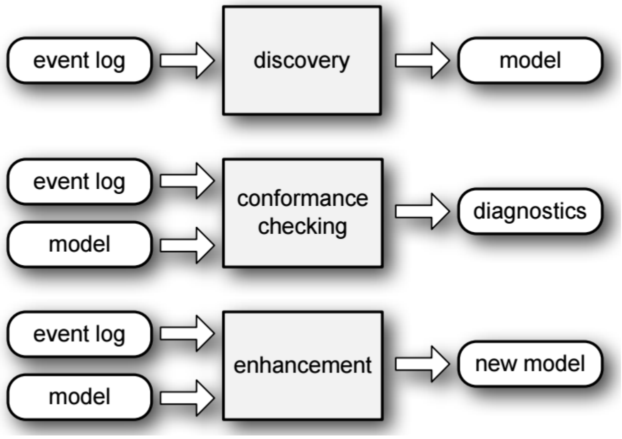


Figure 2. The three basic types of process mining explained in terms of input and output

Orthogonal to the three types of mining, different perspectives can be defined.

- The control-flow perspective. Focuses on the ordering of activities, goal of mining – to find a good characterization of all possible paths (can be expressed in Petri net or some other notation as EPCs, BPMN, UML etc.).
- The organizational perspective. Focuses on the information about resources hidden in the log, goal is to either structure the organization by classifying people in terms of roles and organizational units or to show social network; to structure the organization by classifying people.
- The case perspective. Focuses on properties of cases.
- The time perspective. It's concerned with the timing and frequency of events. It makes possible to discover bottlenecks, measure service levels, monitor the utilization of resources and predict the remaining processing time of running cases.

PM focuses on the relationship between business process models and event data. Inspired by the terminology used by David Harel in the context of Live Sequence Charts [8] there are three types of such relations, which determine the types of analysis.

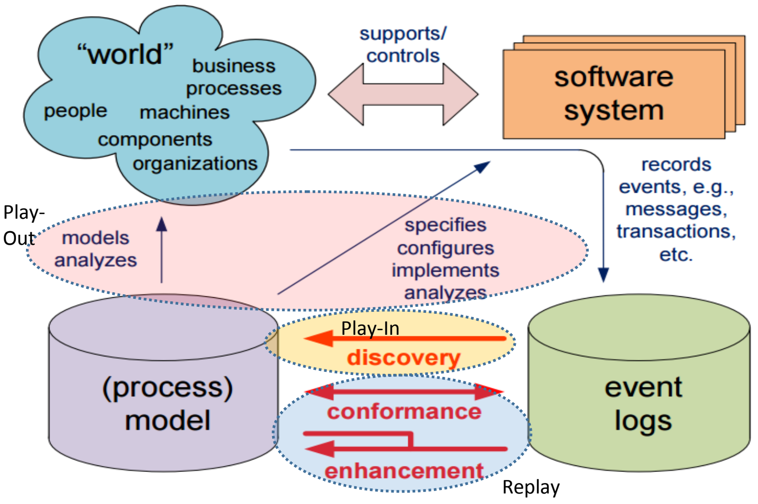
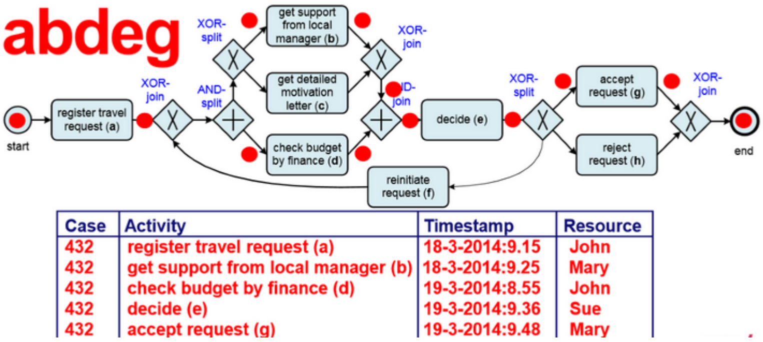


Figure 3. Positioning of the three main types of PM with highlighting areas of relations between a process model and event log

1. Play-Out

Input is finished process model. Next you may simulate different scenarios of a process (according to the model) for filling the event log by data recorded during the simulation events.



Case	Activity	Timestamp	Resource
432	register travel request (a)	18-3-2014:9.15	John
432	get support from local manager (b)	18-3-2014:9.25	Mary
432	check budget by finance (d)	19-3-2014:8.55	John
432	decide (e)	19-3-2014:9.36	Sue
432	accept request (g)	19-3-2014:9.48	Mary

Figure 4. Example of Play-Out.

Above is an example of the finished model to simulate the working process (Workflow). The process model is made BPMN. Red dots show the steps in one of the possible ways to implement the process, and at the bottom of the loge is filled with event data in the order of their registration through the process.

Play-Out is used to validate the developed models of processes for compliance the expected data (sequence of events) with reality.

2. Play-In

It starts with a ready data in the event log. Then get the model of the process, to ensure the implementation presented in the event log (learning process model based on the data).

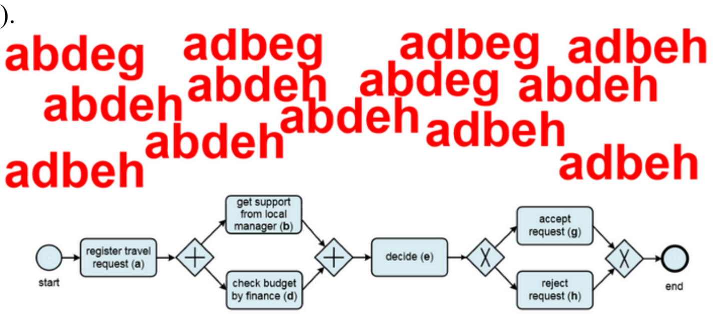


Figure 5. Example of Play-In

All of the sequence of events in the figure above starts with *a* step and end step *g* or *h*. The resulting process model corresponds exactly to the perceived characteristics that illustrates the basic principle of its withdrawal from the data.

Play-In useful for formal description of the processes that generate the known data.

3. Replay

In the figure 6 it’s shown an example of attempts to simulate the existing sequence of events according to the finished model of the process. Attempt failed due to the fact that the model requires that *d* have happened before step *e* (more to deal with the underlying causes of failure will studying of BPMN gateways).

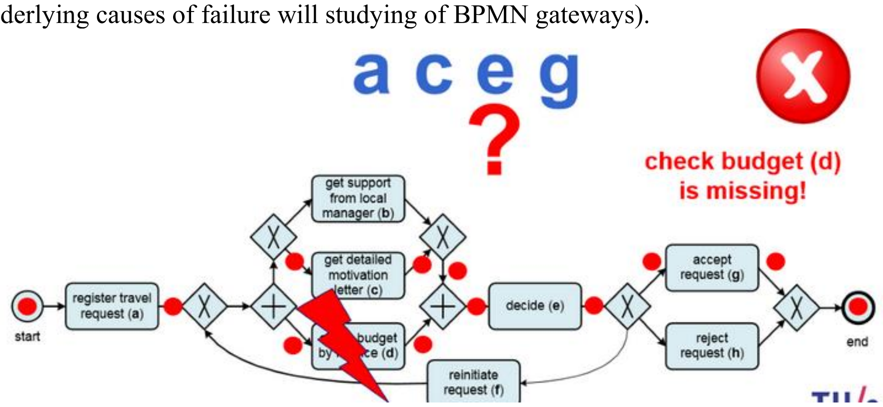

a c e g
?
check budget (d) is missing!
start
register travel request (a)
get support from local manager (b)
get detailed motivation letter (c)
budget by ce (d)
decide (e)
reinitiate request (f)
accept request (g)
reject request (h)
end

Figure 6. Example of Replay

Replay allows to find deviations of models of real processes, but can also be used to analyze the performance of processes.

In short, Play-out generates behavior from existed model and uses for it Petri nets, Workflow, simulation engine and management games. Play-in creates process model from given event log and applies α-algorithm and most data-mining techniques. Finally, Replay has as input both event log and process model and needs for conformance checking, extended the model with frequencies and temporal information, constructing predictive model, operational support. Positioning these three relations is sown in Figure 3.

2.3. Event Log

Event Logs – collection of cases, where each element refers to a case, an activity and a point in time (timestamps).

You can find sources of event data everywhere. For instance, in database system, transaction log (e.g. a trading system), business suite/ ERP system (SAP, Oracle…), message log (e.g. from IBM middleware), open API providing data from websites or social media, CSV (comma-separated values) or spreadsheet etc.

When extracting event log, you can face the next challenges:

1) **Correlation**. Events in an event log are grouped per case. This simple requirement can be quite challenging as it requires event correlation, i.e., events need to be related to each other.

2) **Timestamps**. Events need to be ordered per case. Typical problems: only dates, different clocks, delayed logging.

3) **Snapshots.** Cases may have a lifetime extending beyond the recorded period, e.g. a case was started before the beginning of the event log.

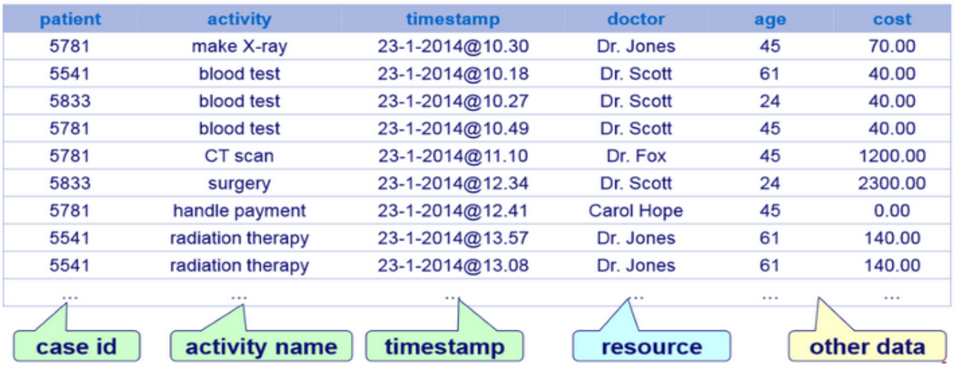
4) **Scoping.** How to decide which tables to incorporate?

5) **Granularity.** The events in the event log are at a different level of granularity than the activities relevant for end users.

Additionally event logs without preprocessing have so called noise and incompleteness. The first one means the event log contains rare and infrequent behavior not representative for the typical behavior of the process. And incompleteness - the event log contains too few events to be able to discover some of the underlying control-flow structures. There are many methods to "clean" data and use only useful data as filtering and data mining techniques.

Every event log must have some certain fields, without that PM will be impossible. Figure 7 clearly shows the basic attributes of the events in the logs:

- Case ID - instances (objects), which are arranged sequence of events log.
- Activity name - actions performed within the event log.
- Timestamp - date and time of recording log events.
- Resource - holds the key actors log events (those who perform actions in the event log).



patient	activity	timestamp	doctor	age	cost
5781	make X-ray	23-1-2014@10.30	Dr. Jones	45	70.00
5541	blood test	23-1-2014@10.18	Dr. Scott	61	40.00
5833	blood test	23-1-2014@10.27	Dr. Scott	24	40.00
5781	blood test	23-1-2014@10.49	Dr. Scott	45	40.00
5781	CT scan	23-1-2014@11.10	Dr. Fox	45	1200.00
5833	surgery	23-1-2014@12.34	Dr. Scott	24	2300.00
5781	handle payment	23-1-2014@12.41	Carol Hope	45	0.00
5541	radiation therapy	23-1-2014@13.57	Dr. Jones	61	140.00
5541	radiation therapy	23-1-2014@13.08	Dr. Jones	61	140.00
...	...	...	...	...	...

Figure 7. Example of Event log (the selection of the attributes depends on the purpose of analysis)

Using Figure 7 we can list some assumptions about event logs:

- A process consists of cases
- A case consists of events such that each event relates to precisely one case.
- Events within a case are ordered.
- Events can have attributes
- Examples of typical attribute names are activity, time, costs, and resource.

Finally, it's worth mentioning some extensions of event logs:

- ✓ Transactional information on activity instance: an event can represent a start, complete, suspend, resume and abort.
- ✓ Case versus event attributes: case attributes don't change, e.g. the birth date or gender, whereas event attributes are related to a particular step in the process.

All process mining techniques assume that it is possible to sequentially record events such that each event refers to an activity (i.e., a well-defined step in some process) and is related to a particular case (i.e., a process instance). Event logs may store additional information about events. In fact, whenever possible, process mining techniques use extra information such as the resource (i.e., person or device) executing or initiating the activity, the timestamp of the event, or data elements recorded with the event (e.g., the size of an order).

2.4. Process Discovery

This type of PM can help to find out what actual process model is. Based just on an event log, a process model is constructed thus capturing the behavior seen in the log.

The most popular algorithms used for this goal are:

- Alpha Miner;
- Alpha+, Alpha++, Alpha#;
- Fuzzy miner;
- Heuristic miner;
- Multi-phase miner;
- Genetic process mining
- Region-based process mining (State-based regions and Language based regions);
- Classical approaches not dealing with concurrency (Inductive inference (Mark Gold, Dana Angluin et al.) and Sequence mining).

Also it's necessary to have a good notification for represent ready process model to end-user. As a rule commonly used Workflow Nets, Petri Nets, Transition Systems, YAWL, BPMN, UML, Causal nets (C-nets) and Event-Driven Process Chain (EPCs).

But any discover technique requires such a representational bios. It helps limiting the search space of possible candidate models. Also it can be used to give preference to particular types of models. It's important to observe process discovery is, by definition, restricted by the expressive power of the target language. Therefore representational bias – the selected target language for presenting and constructing process mining results. Because of every notification isn't universal and has its limitations (e.g. silent steps, work with duplicate activities, with concurrency and loops etc.) and benefits, it recommends to use different variants to correct interpretation of process.

The following are the main characteristics of process discovery algorithms:

1) Representational bias:
    - Inability to represent concurrency
    - Inability to deal with (arbitrary) loops
    - Inability to represent silent actions
    - Inability to represent duplicate actions
    - Inability to model OR-splits/joins
    - Inability to represent non-free-choice behavior
    - Inability to represent hierarchy
2) Ability to deal with noise
3) Completeness notion assumed
4) Used approaches - direct algorithmic approaches (α-algorithm), two-phase approaches (TS, Markov model→WF-net), computational intelligence approaches (genetic algorithm, neural networks, fuzzy sets, swarm intelligence, reinforcement learning, machine learning, rough sets), partial approaches (mining of sequent patterns, discovery of frequent episodes), etc.

2.5. Four quality criteria

Completeness and noise refer to qualities of the event log and do not say much about the quality of the discovered model. m. In fact, there are four competing quality dimensions:

- Fitness. The discovered model should allow for the behavior seen in the event log. A model with good fitness allows for most of the behavior seen in the event log. A model has a perfect fitness if all traces in the log can be replayed by the model from beginning to end.
- Precision. The discovered model should not allow for behavior completely unrelated to what was seen in the event logs (avoid underfitting). Underfitting is the problem that the model overgeneralizes the example behavior in the log (i.e., the model allows for behaviors very different from what was seen in the log).
- Generalization. The discovered model should generalize the example behavior seen in the event logs (avoid overfitting). Overfitting is the problem that a very specific model is generated whereas it is obvious that the log only holds example behavior (i.e., the model explains the particular sample log, but a next sample log of the same process may produce a completely different process model).
- Simplicity. The discovered model should be as simple as possible.

In the Figure 8 it's presented the key criteria for evaluating the quality of process models with appropriate answers that explain meaning of every criteria.

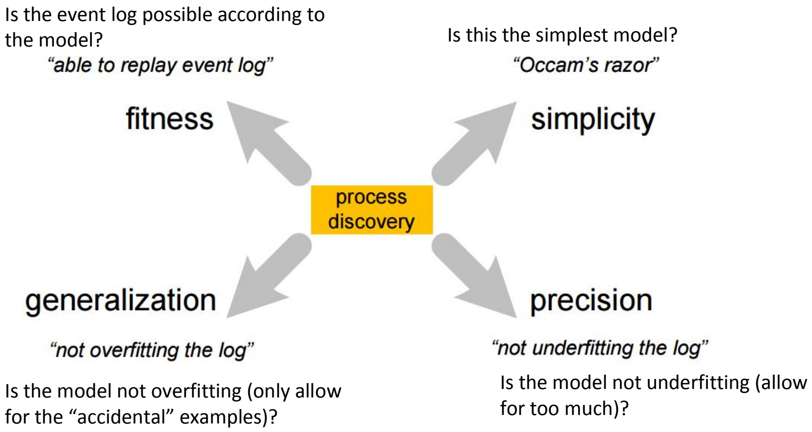

Is the event log possible according to the model?
"able to replay event log"
fitness
Is this the simplest model?
"Occam's razor"
simplicity
process discovery
generalization
"not overfitting the log"
Is the model not overfitting (only allow for the "accidental" examples)?
precision
"not underfitting the log"
Is the model not underfitting (allow for too much)?

Figure 8. The main quality criteria

Balancing fitness, simplicity, precision and generalization is challenging. This is the reason that most of the more powerful process discovery techniques provide various parameters. Improved algorithms need to be developed to better balance the four competing quality dimensions. Moreover, any parameters used should be understandable by end-users.

2.6. Conformance Checking

PM is not limited only by discovery techniques. When one gets appropriate process model, the most interesting and necessary part (for stakeholders) of analyzing begins. The model may have been constructed by hand or may have been discovered. Moreover, the model may be normative or descriptive. Conformance checking relates in the event log to activities in the process model and compares both. It needs an event log and a model as input. The goal is to find commonalities and discrepancies between the modeled behavior and the observed behavior. This type of PM is relevant for business alignment and auditing. The different types of models can be considered: conformance checking can be applied to procedural models, organizational models, declarative process models, business rules/policies, laws, etc.

Generally, conformance checking is used for:

- improving the alignment of business processes, information systems and organizations;
- auditors;
- repairing models;
- evaluating process discovery algorithm;
- connecting event log and process model.

Following description of the technique for conformance checking will be mainly focused on the one of quality criteria – fitness, because the other three quality criteria are less relevant.

2.6.1. Conformance checking using Token-based play

The idea of this method is counting tokens while replaying, i.e. to simply count the fraction of cases that can be "parsed completely" (the proportion of cases corresponding to firing sequences leading from <start> to <end>). While replaying on top of, for example, WF-net, we have four counters: *p* (produced tokens), *c* (consumed tokens), *m* (missing tokens – consumed while not there) and *r* (remaining tokens – produced but not consumed). Initially, p=c=0, then the environment produces a token for place <start>. At the end, the environment consumes a token from place end. For instance, if we try replay trace 6= <a,d,c,e,h> on the top of given model, then final state of replay will look like in the figure 9. For this case, we will have following counters: p=6, c=6, m=1, r=1.

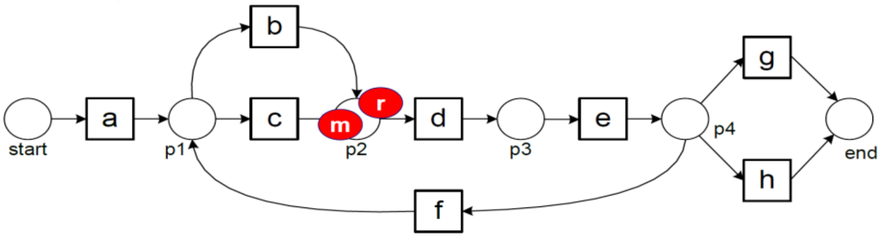


Figure 9. Replaying trace 6= < a,d,c,e,h > on top of WF-net (final state)

Fitness for trace will be computed by formula:

$$fitness(6\,,N) = \frac{1}{2}\left(1-\frac{m}{c}\right)+\frac{1}{2}(1-\frac{r}{p}) \qquad (1)$$

,where 6– trace from event log, N – process model.

Then fitness for trace 6will be equal to:

$$fitness(6,N) = \frac{1}{2}\left(1-\frac{1}{6}\right)+\frac{1}{2}\left(1-\frac{1}{6}\right) = 0.8333 \qquad (2)$$

The fitness of an event log L on WF-net N is defined as:

$$fitness(\mathrm{L},N) = \frac{1}{2}\left(1-\frac{\sum_{6\in \mathrm{L}} L(6)*m_{N,6}}{\sum_{6\in \mathrm{L}} L(6)*c_{N,6}}\right)+\frac{1}{2}(1-\frac{\sum_{6\in \mathrm{L}} L(6)*r_{N,6}}{\sum_{6\in \mathrm{L}} L(6)*p_{N,6}}) \qquad (3)$$

By help this method we can analyze and detect problem in compliance as it's shown in figure below.

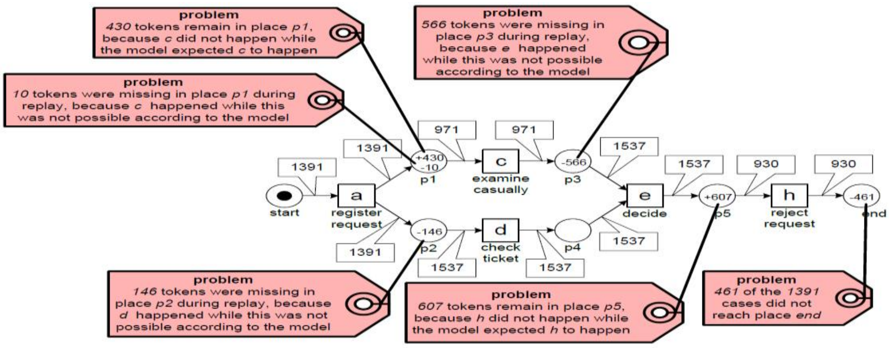
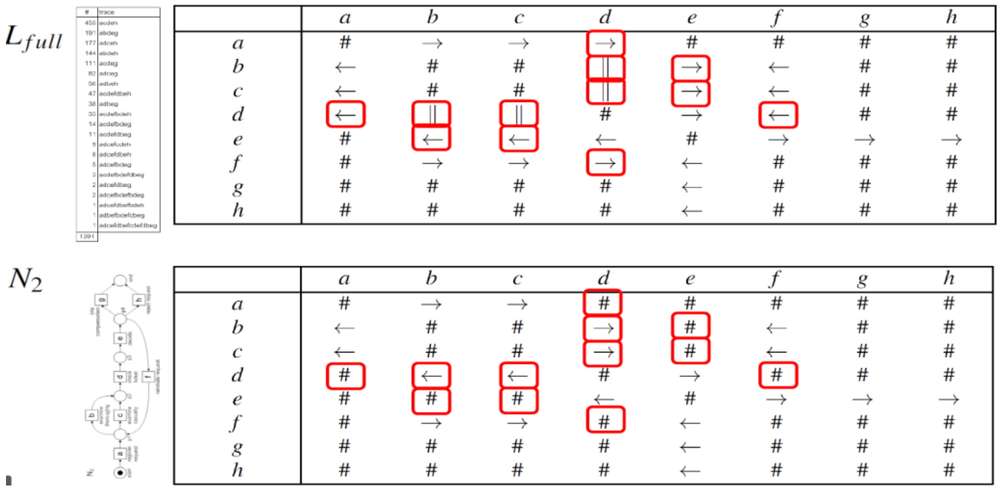

problem
430 tokens remain in place p1, because c did not happen while the model expected c to happen
problem
566 tokens were missing in place p3 during replay, because e happened while this was not possible according to the model
problem
10 tokens were missing in place p1 during replay, because c happened while this was not possible according to the model
971
971
1391
1537
1391
+430
-10
p1
c
examine casually
-566
p3
1537
930
930
start
a
register request
e
decide
+607
p5
h
reject request
-461
end
-146
p2
d
check ticket
p4
1537
1391
1537
1537
problem
146 tokens were missing in place p2 during replay, because d happened while this was not possible according to the model
problem
607 tokens remain in place p5, because h did not happen while the model expected h to happen
problem
461 of the 1391 cases did not reach place end

Figure 10. Detection of problems by token-based play.

2.6.2. Conformance checking using causal footprints

Conformance analysis based on footprints is only meaningful if the log is complete with respect to the “directly follows” relation. By counting differencies (viz. Figure 10) we can compte fitness. Footprint – matrix showing causal dependencies such as:

- Direct succession: x>y iff for some case x is directly followed by y.
- Causality: x→y iff x>y and not y>x.
- Parallel: x||y iff x>y and y>x
- Choice: x#y iff not x>y and not y>x.



L_{full}

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
a	#	→	→	→	#	#	#	#
b	←	#	#	||	→	←	#	#
c	←	#	#	||	→	←	#	#
d	←	||	||	#	→	←	#	#
e	#	←	←	←	#	→	→	→
f	#	→	→	→	←	#	#	#
g	#	#	#	#	←	#	#	#
h	#	#	#	#	←	#	#	#

N_2

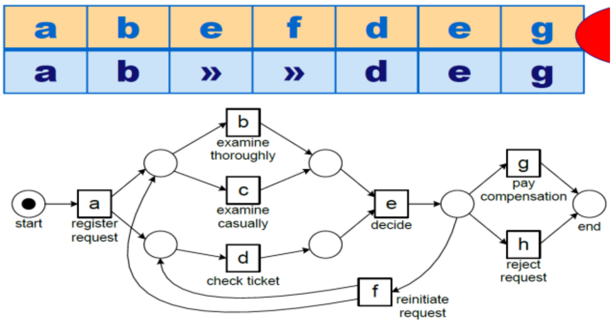
	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
a	#	→	→	#	#	#	#	#
b	←	#	#	→	#	←	#	#
c	←	#	#	→	#	←	#	#
d	#	←	←	#	→	#	#	#
e	#	#	#	←	#	→	→	→
f	#	→	→	#	←	#	#	#
g	#	#	#	#	←	#	#	#
h	#	#	#	#	←	#	#	#

Figure 11. Differences between the footprints of L_{full} and N_2.

This method allows for log-to-model comparisons, i.e. it can be checked whether and model and event log “agree” on the ordering of activities. However, the same approach can be used for log-to-log and model-to-model comparisons.

2.6.3. Conformance checking using alignment

It should provide a "closest matching path" through the process model for any trace in the event log. Also required for performance analysis. In figure 12 the first string shows possible trace in model and second one shows move in log only. Cost of alignment are differences between model and log. The goal is to find the most optimal alignment (with minimal cost).



a	b	e	f	d	e	g
a	b	»	»	d	e	g

Figure 12. Aligning event log and process model.

2.7. Model enhancement.

A process model is extended or improved using information extracted from some log. As seen before, event logs contain much more information that goes far beyond just control-flow, namely information about resources, time, and data attributes etc. Organizational mining can be used to get insight into typical work patterns, organizational structures, and social networks. Timestamps and frequencies of activities can be used to identify bottlenecks and diagnose other performance related problems. Case data can be used to better understand decision-making and analyze differences among cases. The different perspectives can be merged into a single integrated process model for next simulation and “what if” analysis to explore different redesigns and control strategies. In the figure 13 it’s presented approach to come to a fully integrated model covering the organizational, time, and case perspectives.

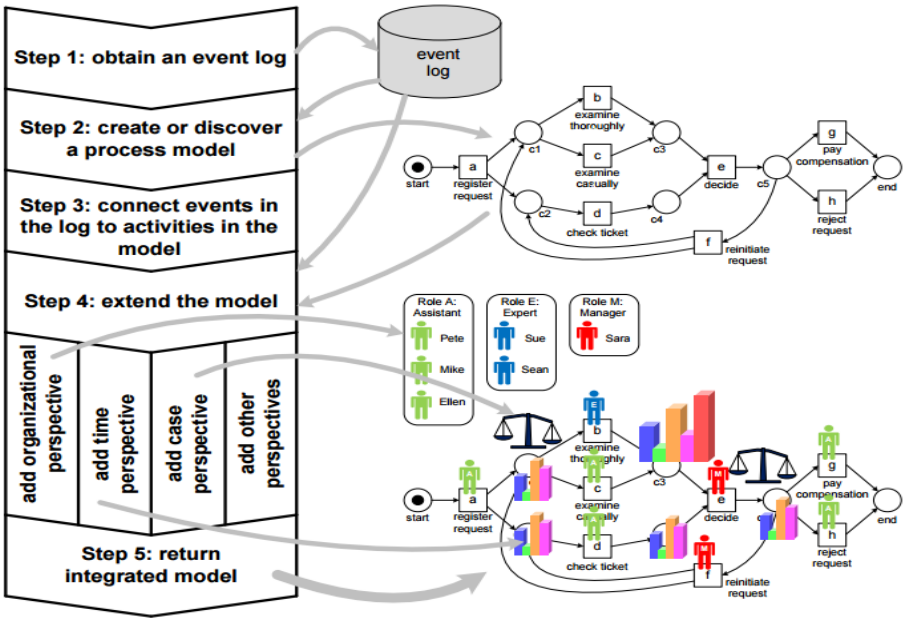


Figure 13. Connecting event log and model (with extension)

2.8. Refined Process mining Framework

Today many data are updated in real-time and sufficient computing power is available to analysis events when they occur. Therefore, PM should not be restricted to off-line analysis and can also be used for online operational support.

Figure 14 shows refined PM Framework (can be extended). Provenance refers to the data that is needed to be able to reproduce an experiment. Data in event logs are portioned into “pre mortem” and “post mortem”. “”Post mortem” – information about cases that have completed and can be used for process improvement and auditing, but not for influencing the cases. “Pre mortem” – cases that have not yet completed and can be exploited to ensure the correct or efficient handling the cases.

The refined PM Framework also distinguish between two types of models: “de jure models” and “de facto models”. The first one is normative, and second one is descriptive.

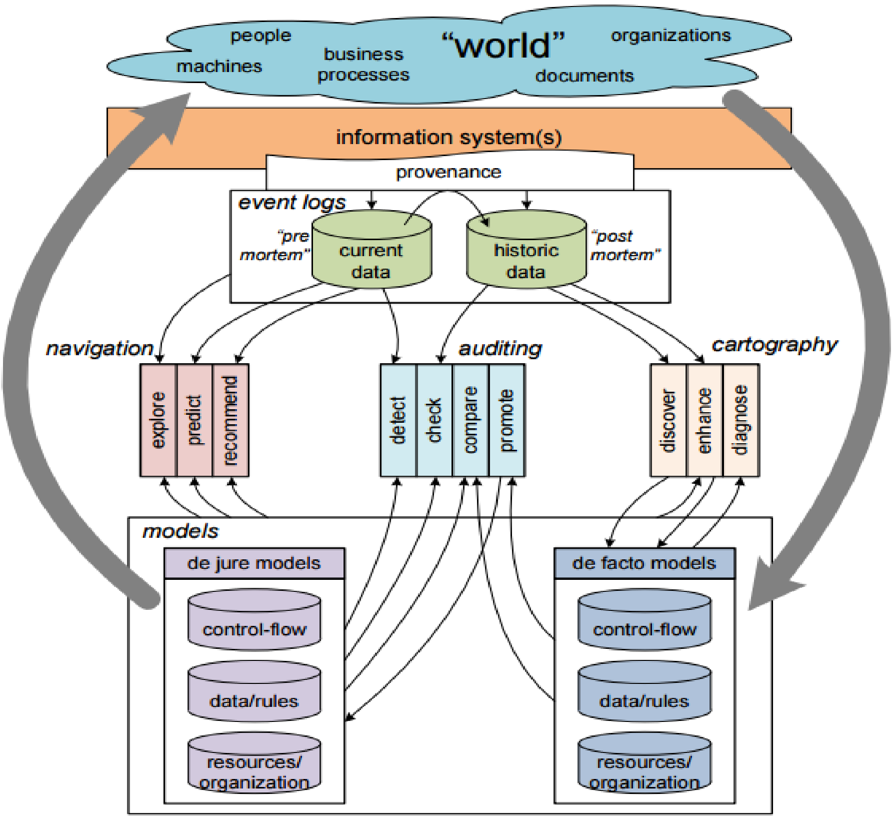


Figure 14. Refined PM Framework

So let’s consider category of PM activities more detailed.

PM can be seen as the “maps” describing the operational processes of organizations. Group Cartography includes three activities:

- Discover. This activity is concerned with the extraction of (process) models.
- Enhance. When existing process models (either discovered or hand-made) can be related to events logs, it is possible to enhance (extend and repair) these models.
- Diagnose. This activity does not directly use event logs and focuses on classical model-based analysis.

Group Auditing obtains set of activities used to check whether the business processes are executed within certain boundaries set by managers, governments, and other stakeholders.

- Detect. Compares de jure models with current “pre mortem” data. The moment a predefined rule is violated, an alert is generated (online).
- Check. The goal of this activity is to pinpoint deviations and quantify the level of compliance (offline).
- Compare. De facto models can be compared with de jure models to see in what way reality deviates from what was planned or expected.
- Promote. Promote parts of the de facto model to a new de jure model.

And last category is Navigation. It’s forwardlooking, helps in supporting and guiding process execution (unlike the Cartography and Auditing).

- Explore. The combination of event data and models can be used to explore business processes at run-time.
- Predict. By combining information about running cases with models, it is possible to make predictions about the future, e.g., the remaining flow time and the probability of success.
- Recommend. The information used for predicting the future can also be used to recommend suitable actions (e.g. to minimize costs or time).

In next part this category will be described more detailed, because of it can be used for online analysis and influencing on current process.

2.9. Operation support

2.9.1 Detect

Figure below illustrates type of operational support. Users are interacting with some enterprise information system. Based on their actions, event are recorded. The partial trace of each case is continuously checked by the operational support system, which immediately generates an alert if a deviation is detected.

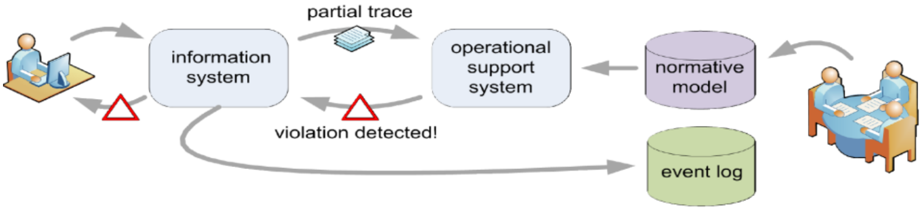


Figure 15. Detecting violations at run-time.

2.9.2 Predict

We again consider the setting in which users are interacting with some enterprise information system (viz. Figure 16). The events recorded for cases can be sent to the operational support system in the form of partial traces. Based on such a partial trace and some predictive model, a prediction is generated.

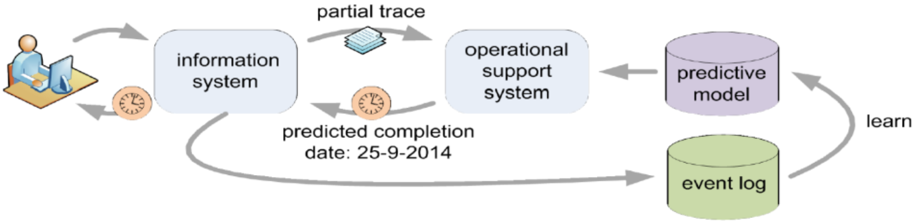

partial trace
information system
operational support system
predictive model
learn
predicted completion date: 25-9-2014
event log

Figure 16. Both the partial trace of a running case and some predictive model are used to provide a prediction.

2.9.3. Recommend

The setting is similar to prediction. However, the response is not a prediction but a recommendation about what do next (viz. Figure 17). To provide such a recommendation, a model is learned from "post mortem" data. A recommendation is always given with respect to a specific goal. For example, to minimize the remaining flow time or the total cost, to maximize the fraction of cases handled within 4 weeks, etc.

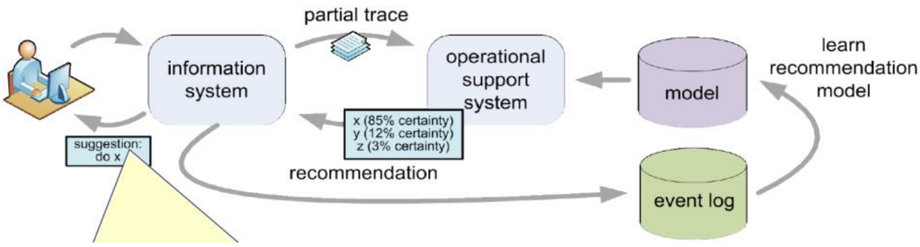


Figure 17. A model based on historic data is used to provide recommendations for running cases.

2.10. Tools

All techniques described above were realized in such software as PROM. ProM is an extensible framework that supports a wide variety of process mining techniques in the form of plug-ins.

The main characteristics of PROM:

- Aims to cover the whole process mining spectrum.
- Notations supported: Petri nets (many types), BPMN, C-nets, fuzzy models, transition systems, Declare, etc.
- Also supports conformance checking and operational support.
- Many plug-ins are experimental prototypes and not user friendly.

This is extremely powerful instrument, but confusing for someone. Nowadays already exist 600 plug-ins and this amount grows up.

There is also commercial software Disco that has following characteristics:

- Focus on discovery and performance analysis (including animation).
- Powerful filtering capabilities for comparative process mining and ad-hoc checking of patterns.
- Uses a variant of fuzzy models, etc.
- Does not support conformance checking and operational support.
- Easy to use and excellent performance.

Disco can be used by unexperienced people, has intuitive user-friendly interface.

Tools are available (Figure 18), but process mining is still a relatively young discipline. New tools will appear in coming years and process mining functionality will be embedded in more BI/BPM/DM suites.

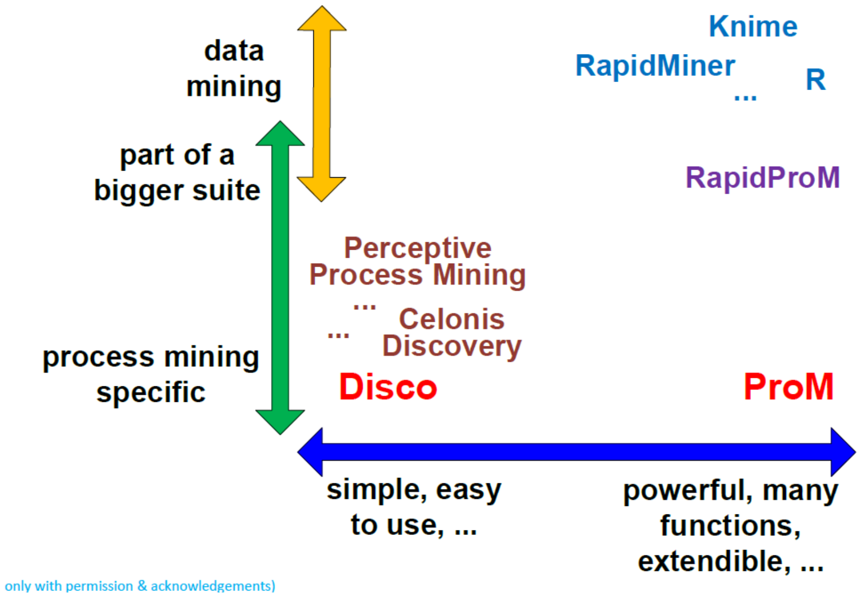


Figure 18. Spectrum of PM tools

In table 2 described popular tools for process mining with specifying benefits and lacks.

Product name	Famous advantages	Famous disadvantages
Reflect|One	Supported BPM life-cycle, reflect user-friendliness, scalability, support organizational mining by social networks, Discovery algorithms: on a genetic mining, on a sequential model.	Not support conformance checking and prediction
Disco	Focus on high performance, support seamless abstraction and generalization using the cartography, deal with complex (Spaghetti-like) processes, have Nitro, that can recognize different time formats and automatically maps these onto XES or MXML notation Algorithm based on the fuzzy mining.	
Enterprise Visualization Suite	Focus on analysis of SAP supported business processes Process discovery algorithm inspired by α algorithm and heuristic mining	

Product name	Famous advantages	Famous disadvantages
InterStage BPME	Not need to install, focus on process discovery, able to seamlessly abstract from infrequent behavior, able to analyze performance using indicators such flow time	Unable to discover concurrency, not support prediction, recommendation and conformance checking
ARIS Process Performance Manager	Focus on performance analysis (drilling down to the instance level, benchmarking, dashboards), support organizational mining	Not support prediction, recommendation and conformance checking
Genet/Petrify Rbminer/Dbminer	Use state/based regions, support control-flow discovery, rely on Prom for conformance checking	
Service Mosaic	Analysis of service interaction logs, discovers transition systems, focus on dealing with noise and protocol refinement	Unable to discover concurrency

Table 2. Example of process mining products.

2.11. Types of processes

2.11.1. Lasagna Processes

Lasagna processes are relatively structured and the cases flowing through such processes are handled in a controlled manner. Therefore it’s possible to apply all of th PM techniques presented in the preceding chapters. Definition of Lasagna process is: a process is a Lasagna process if with limited efforts it is possible to create an agreed-upon process model that has a fitness of at least 0.8, i.e., more than 80% of the events happen as planned and stakeholders confirm the validity of the model. In figure below presented example of Lasagna process by help Petri net and BPMN.

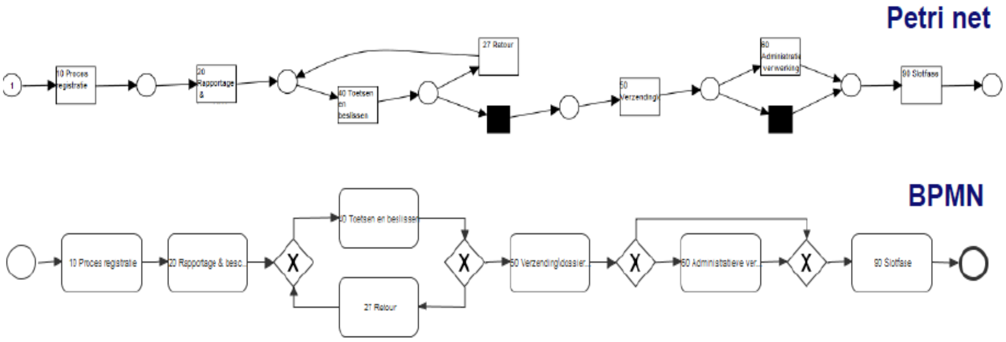


Figure 19. Example of usual Lasagna process

So the main characteristics of Lasagna processes are:

- Easy to discover, but it is less interesting to show the "real" process. (close to expectation)
- Whole process mining toolbox can be applied.
- Added value is predominantly in more advanced forms of process mining based on aligning log and model.

2.11.2. Spaghetti Processes

Spaghetti processes are less structured than Lasagna processes, only some of process mining techniques can be applied. Figure below shows why unstructured processes are called Spaghetti processes. There are different approaches to get valuable analyze from such kind of processes. For example, method Divide and Conquer (by clustering of cases) or showing only the most frequencies paths and activities (Disco).

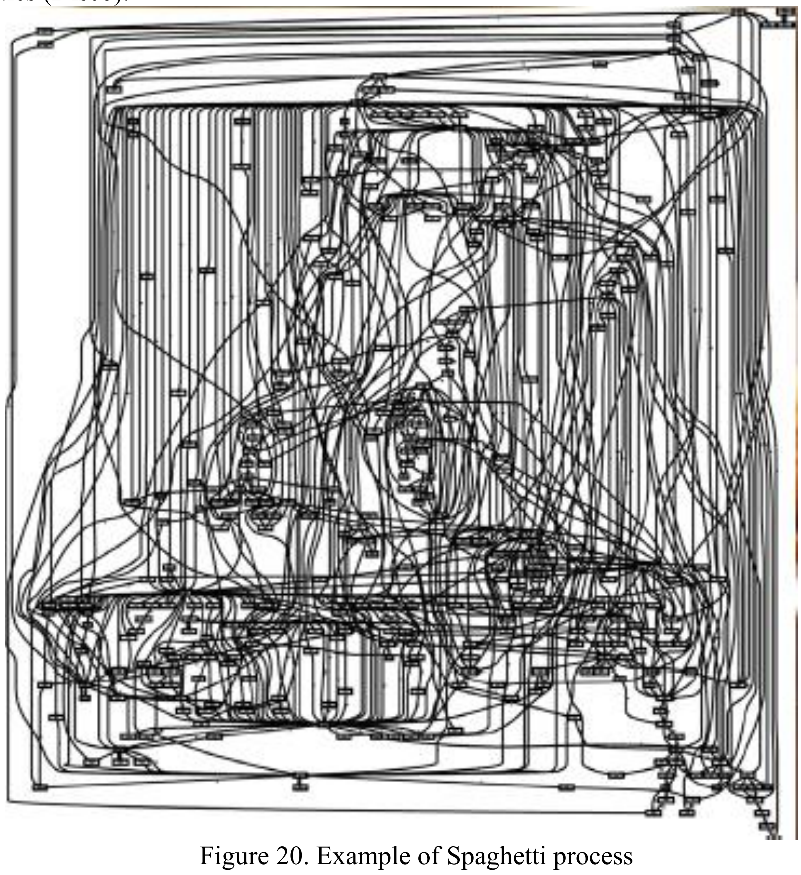


Figure 20. Example of Spaghetti process

2.11.3. Applications of both types of model

In figure 21 depicted overview of the different functional areas in a typical organization. Lasagna processes are typically encountered in production, finance/accounting, procurement, logistics, resource management, and sales/CRM. Spaghetti processes are typically encountered in product development, service, resource management, and sales/CRM.

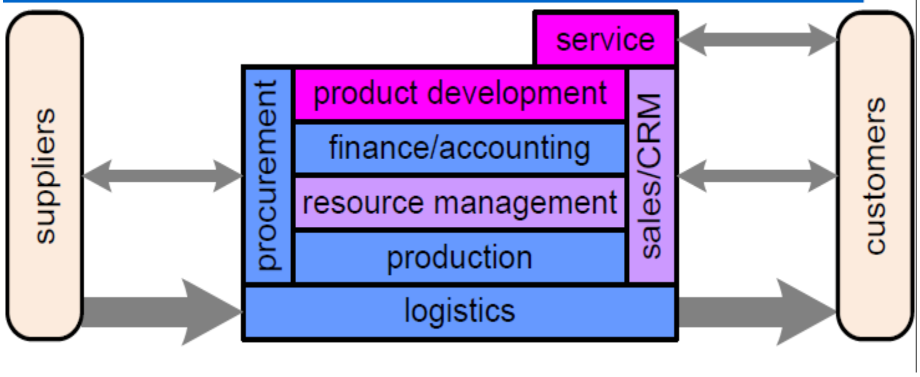


Figure 21. Applications of Spaghetti (violet cells), Lasagna (blue cells) processes and both (pink cells).

Nevertheless, Spaghetti processes are very interesting from the viewpoint of PM as they often allow for various improvements. A highly-structured well-organized process is often less interesting in this respect; it's easy to apply PM techniques but there is also little improvement potential.

3. Conclusions

PM is important tool for modern organizations that need to manage nontrivial operational processes. Data mining techniques aim to describe and understand reality based on historic data, but it’s low level of analyze, because this techniques are nor process-centric. Unlike most BPM approaches, PM is driven by factual event data rather than hand-made models. That’s why PM is called a bridge between BPM and Data Mining.

PM is not limited to process discover. By connecting event log and process model, new ways for analyzing are opened. Discovered process model can be also extended by information from various perspectives.

Nevertheless, as enough new approach PM has a lot of unsolved challenges. Some of them are following:

- There are no negative examples (i.e., a log shows what has happened but does not show what could not happen).
- Preprocessing of Event log (problems with Noise and Incompleteness)
- Concept drift [13, 14]
- There is no clear how to correct recognize attributes of event log[12]
- Due to concurrency, loops, and choices the search space has a complex structure and the log typically contains only a fraction of all possible behaviors.
- There is no clear relation between the size of a model and its behavior (i.e., a smaller model may generate more or less behavior although classical analysis and evaluation methods typically assume some monotonicity property).
- Improving the Representational Bias Used for Process Discovery
- Balancing Between Quality Criteria such as Fitness, Simplicity, Precision, and Generalization
- Improving Usability and Understandability for NonExperts
- Cross-Organizational Mining
- etc.

Moreover, PM can be used off-line and online. And from a technological point of view online PM may be challenged.

Even so mature PM techniques and tools are available and it successfully used in over organizations.

References

[1] Wil M.P. van der Aalst Process Mining: Discovery, Conformance and Enhancement of Business Processes

ISBN: 978-3-642-19344-6, DOI 10.1007/978-3-642-19345-3, Springer Heidelberg Dordrecht London New York, 2002

[2] Massive Open Online Course: "Process Mining: Data science in Action". Wil van der Aalst Eindhoven University of Technology, Department Methematics & Computer Science

[3] IEEE CIS Task Force on Process Mining. Process Mining Manifesto

[4] Wil van der Aalst :“Process Mining: X-Ray Your Business Processes“, Process Mining, Communications of the ACM CACM Volume 55 Issue 8 (August 2012) Pages 76- 83, <http://doi.acm.org/10.1145/2240236.2240 257>.

[5] Anne Rozinat1, Wil van der Aalst Process Mining: The Objectification of Gut Instinct - Making Business Processes More Transparent Through Data Analysis

[6] Website www.habrahabr.ru/post/244879/, article Introduction in Process Mining

[7] Michael Palmer Data is the New Oil

[8] D.Harel and R.Marelly. Come, Let’s play: Scenari-Based Programming Using LSCs and Play-Engine. Springer, Berlin, 2003.

[9] Website www.processmining.org

[10] Website www.fluxicon.com

[11]J.C. Bose, R. Mans, and W. van der Aalst. Wanna improve process mining results? Computational Intelligence and Data Mining (CIDM 2013), doi: 10.1109/CIDM.2013.6597227

[12] Burattin Andrea. Applicability of Process Mining techniques in Business Environments

[13]Christian Günter. Process Mining in Flexible Environment.

[14] Arjel Bautista, Lalit Wangikar, S.M. Kumail Akbar “Process Mining-Driven Optimization of a Consumer Loan Approvals Process” CKM Advisors, 711 Third Avenue, Suite 1806, NY, USA