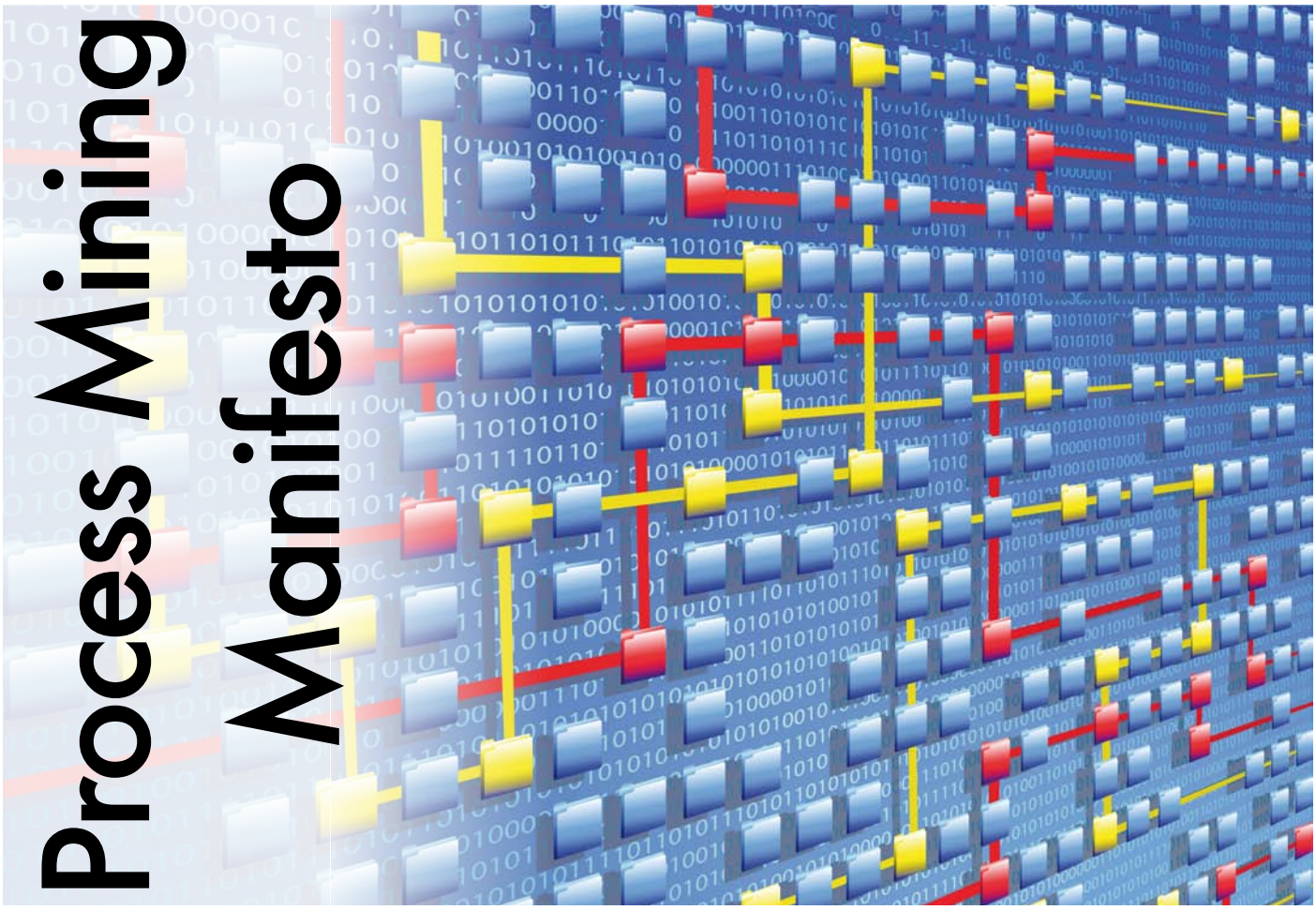
Process Mining Manifesto



A *manifesto* is a "public declaration of principles and intentions" by a group of people. This manifesto is written by members and supporters of the *IEEE Task Force on Process Mining*. The goal of this task force is to promote the research, development, education, implementation, evolution, and understanding of process mining.

Process mining is a relatively young research discipline that sits between computational intelligence and data mining on the one hand, and process modeling and analysis on the other hand. The idea of process mining is to *discover, monitor and improve real processes* (i.e., not assumed processes) *by extracting knowledge from event logs* readily available in today's (information) systems. Process mining 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.

Contents:

Process mining 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. On the one hand, more and more events are being recorded, thus, providing detailed information about the history of processes. On the other hand, there is a need to improve and support business processes in competitive and rapidly changing environments. This manifesto is created by the *IEEE Task Force on Process Mining* and aims to promote the topic of process mining. Moreover, by defining a set of guiding principles and listing important challenges, this manifesto hopes to serve as a *guide for software developers*, *scientists*, *consultants*, *business managers*, and *end-users*. The goal is to increase the maturity of process mining as a new tool to improve the (re)design, control, and support of operational business processes.

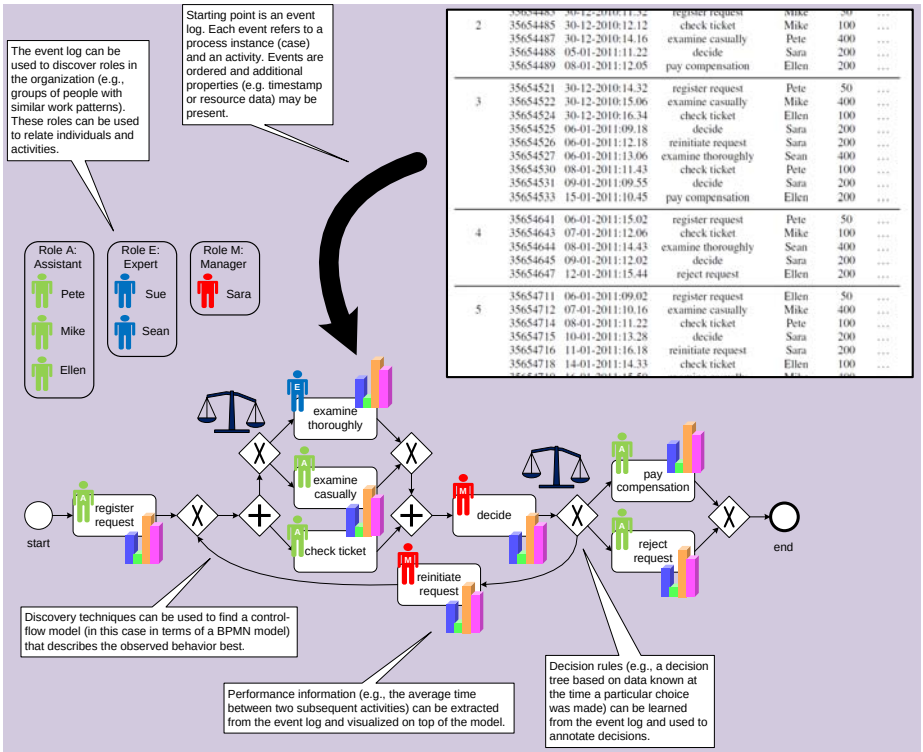


Figure 1: Process mining techniques extract knowledge from event logs in order to discover, monitor and improve processes.

Process mining provides an important bridge between data mining and business process modeling and analysis. Under the *Business Intelligence* (BI) umbrella many buzzwords have been introduced to refer to rather simple reporting and dashboard tools. *Business Activity Monitoring* (BAM) refers to technologies enabling the real-time monitoring of business processes. *Complex Event Processing* (CEP) refers to technologies to process large amounts of events, utilizing them to monitor, steer and optimize the business in real time. *Corporate Performance Management* (CPM) is another buzzword for measuring the performance of a process or organization. Also related are management approaches such as *Continuous Process Improvement* (CPI), *Business Process Improvement* (BPI), *Total Quality Management* (TQM), and *Six Sigma*. These approaches have in common that processes are "put under a microscope" to see whether further improvements are possible. Process mining is an enabling technology for CPM, BPI, TQM, Six Sigma, and the like.

Whereas BI tools and management approaches such as Six Sigma and TQM aim to improve operational performance, e.g., reducing flow time and defects, organizations are also putting more emphasis on *corporate governance*, *risks*, and *compliance*. Legislations such as the Sarbanes-Oxley Act (SOX) and the Basel II Accord illustrate the focus on compliance issues. Process mining techniques offer a means to more rigorously check compliance and ascertain the validity and reliability of information about an organization's core processes.

Over the last decade, event data have become readily available and process mining techniques have matured. Moreover, as just mentioned, management trends related to process improvement (e.g., Six Sigma, TQM, CPI, and CPM) and compliance (SOX, BAM, etc.) can benefit from process mining. Fortunately, process mining algorithms have been implemented in various academic and commercial systems. Today, there is an active group of researchers working on process mining and it has become one of the "hot topics" in Business Process Management (BPM) research. Moreover, there is a huge interest from industry in process mining. More and more software vendors are adding process mining functionality to their tools. Examples of software products with process mining capabilities are: ARIS Process Performance Manager (Software AG), Comprehend (Open Connect), Discovery Analyst (StereoLOGIC), Flow (Fourspark), Futura Reflect (Futura Process Intelligence), Interstage Automated Process Discovery (Fujitsu), OKT Process Mining suite (Exeura), Process Discovery Focus (Iontas/Verint), ProcessAnalyzer (QPR), ProM (TU/e), Rbminer/Dbminer (UPC), and Reflect| one (Pallas Athena). The growing interest in log-based process analysis motivated the establishment of a Task Force on Process Mining.

The task force was established in 2009 in the context of the Data Mining Technical Committee (DMTC) of the Computational Intelligence Society (CIS) of the Institute of Electrical and Electronic Engineers (IEEE). The current task force has members representing *software vendors* (e.g., Pallas Athena, Software AG, Futura Process Intelligence, HP, IBM, Infosys, Fluxicon, Businesscape, Iontas/Verint, Fujitsu, Fujitsu Laboratories, Business Process Mining, Stereologic), *consultancy firms/ end users* (e.g., ProcessGold, Business Process Trends, Gartner, Deloitte, Process Sphere, Siav SpA, BPM Chile, BWI Systeme GmbH, Excellentia BPM, Rabobank), and *research institutes* (e.g., TU/e, University of Padua, Universitat Politècnica de Catalunya, New Mexico State University, IST - Technical University of Lisbon, University of Calabria, Penn State University, University of Bari, Humboldt-Universität zu Berlin, Queensland University of Technology, Vienna University of Economics and Business,

Concrete objectives of the task force:

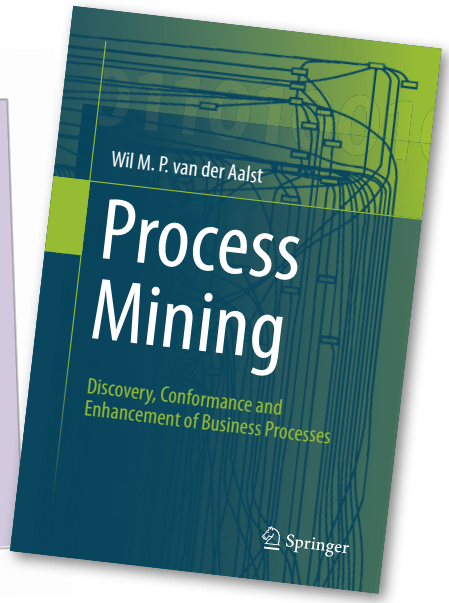
1) Make end-users, developers, consultants, business managers, and researchers aware of the state-of-the-art in process mining.
2) Promote the use of process mining techniques and tools and stimulate new applications.
3) Play a role in standardization efforts for logging event data.
4) Organize tutorials, special sessions, workshops, panels.
5) Publish articles, books, videos, and special issues of journals.

Stevens Institute of Technology, University of Haifa, University of Bologna, Ulsan National Institute of Science and Technology, Cranfield University, K.U. Leuven, Tsinghua University, University of Innsbruck, University of Tartu).

Since its establishment in 2009 there have been various activities related to the above objectives. For example, several workshops and special tracks were (co-)organized by the task force, e.g., the workshops on Business Process Intelligence (BPI'09, BPI'10, and BPI'11) and special tracks at main IEEE conferences (e.g. CIDM'11). Knowledge was disseminated via tutorials (e.g. WCCI'10 and PMPM'09), summer schools (ESSCaSS'09, ACPN'10, CICH'10, etc.), videos (cf. www.processmining.org), and several publications including the first book on process mining recently published by Springer. The task force also (co-)organized the first Business Process Intelligence Challenge (BPIC'11): a competition where participants had to extract meaningful knowledge from a large and complex event log. In 2010, the task force also standardized *XES* (www.xes-standard.org), a standard logging format that is extensible and supported by the *OpenXES library* (www.openxes.org) and by tools such as ProM, XESame, Nitro, etc.

Process Mining Book

www.processmining.org/book/
W.M.P. van der Aalst. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, Berlin, 2011.



The reader is invited to visit www.win.tue.nl/ieeetfpm/ for more information about the activities of the task force.

2. Process Mining: State of the Art

The expanding capabilities of information systems and other systems that depend on computing, are well characterized by Moore's law. Gordon Moore, the co-founder of Intel, predicted in 1965 that the number of components in integrated circuits would double every year. During the last fifty years the growth has indeed been exponential, albeit at a slightly slower pace. These advancements resulted in a spectacular growth of the "digital universe" (i.e., all data stored and/or exchanged electronically). Moreover, the digital and the real universe continue to become more and more aligned.

The growth of a digital universe that is well-aligned with processes in organizations makes it possible to record and analyze *events*. Events may range from the withdrawal of cash from an ATM, a doctor adjusting an X-ray machine, a citizen applying for a driver license, the submission of a tax declaration, and the receipt of an e-ticket number by a traveler. The challenge is to exploit event data in a meaningful way, for example, to provide insights, identify bottlenecks, anticipate problems, record policy violations, recommend countermeasures, and streamline processes. Process mining aims to do exactly that.

Starting point for process mining is an *event log*. 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).

As shown in Fig. 2, event logs can be used to conduct three types of process mining. The first type of process mining is *discovery*. A discovery technique takes an event log and

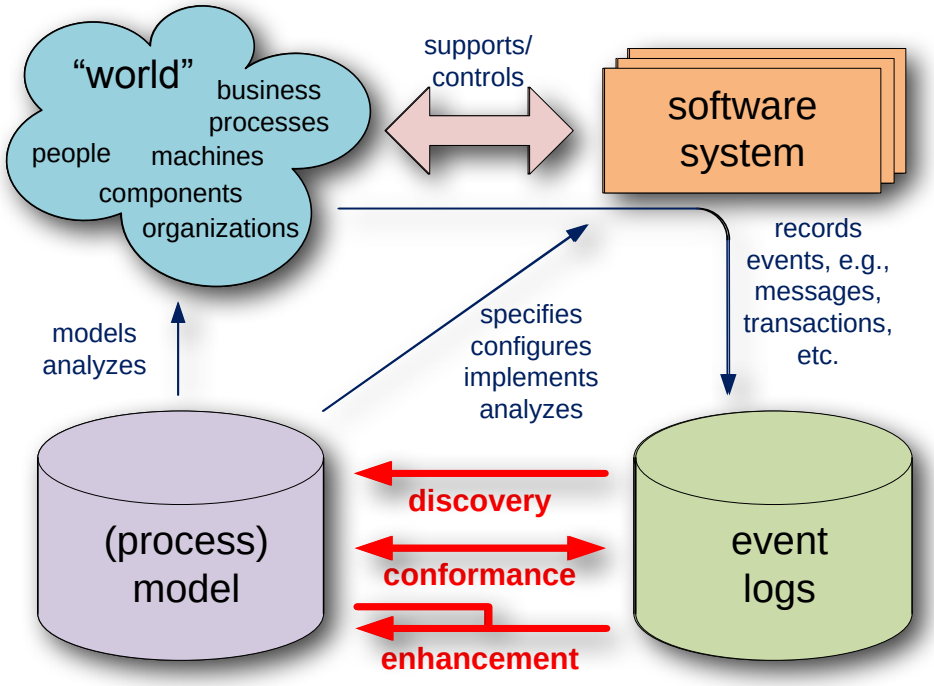


Figure 2: Positioning of the three main types of process mining: (a) *discovery*, (b) *conformance* checking, and (c) *enhancement*.

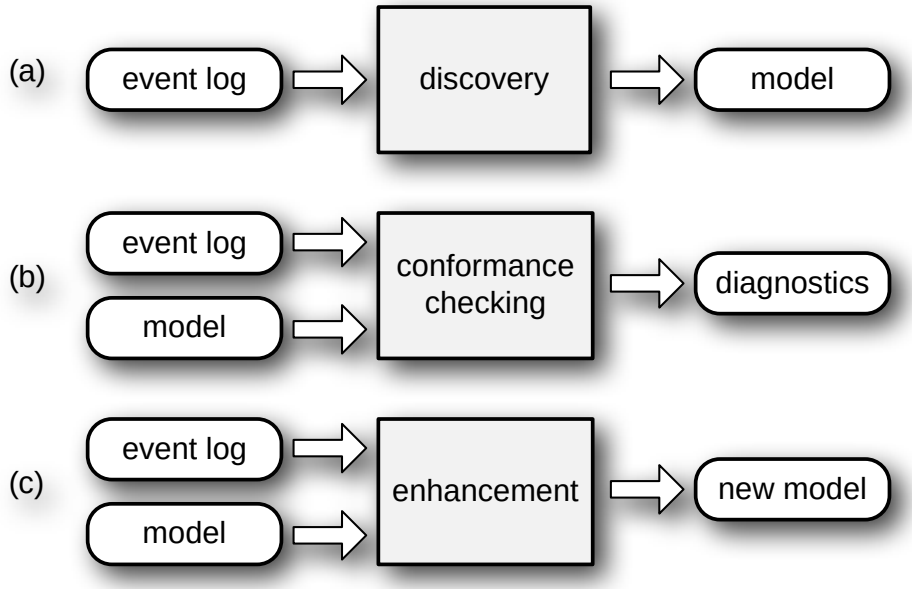


Figure 3: The three basic types of process mining explained in terms of input and output: (a) discovery, (b) conformance checking, and (c) enhancement.

produces a model without using any a-priori information. Process discovery is the most prominent process mining technique. For many organizations it is surprising to see that existing techniques are indeed able to discover real processes merely based on example executions in event logs. 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. Note that 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. The third type of process mining is *enhancement*. Here, the 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. For instance, by using timestamps in the event log one can extend the model to show bottlenecks, service levels, throughput times, and frequencies.

Figure 3 describes the three types of process mining in terms of input and output. Techniques for discovery take an event log and produce a model. The discovered model is typically a process model (e.g., a Petri net, BPMN, EPC, or UML activity diagram), however, the model may also describe other perspectives (e.g., a social network). Conformance checking techniques need an event log and a model as input. The output consists of diagnostic information showing differences and commonalities between model and log. Techniques for model enhancement (repair or extension) also need an event log and a model as input. The output is an improved or extended model.

Process mining may cover different perspectives. The *control-flow perspective* focuses on the control-flow, i.e., the ordering of activities. The goal of mining this perspective is to find a good characterization of all possible paths. The result is typically expressed in terms of a Petri net or some other process notation (e.g., EPCs, BPMN, or UML activity diagrams). The *organizational perspective* focuses on information about resources hidden in the log, i.e., which actors (e.g., people, systems, roles, or departments) are involved and how are they related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or to show the social network. The *case perspective* focuses on properties of cases. Obviously, a case can be characterized by its path in the process or by the actors working on it. However, cases can also be characterized by the values of the corresponding data elements. For example, if a case represents a

Process Mining Characteristics:

1. Process mining is not limited to control-flow discovery.
The discovery of process models from event logs fuels the imagination of both practitioners and academics. Therefore, control-flow discovery is often seen as the most exciting part of process mining. However, process mining is not limited to control-flow discovery. On the one hand, discovery is just one of the three basic forms of process mining (discovery, conformance, and enhancement). On the other hand, the scope is not limited to control-flow; the organizational, case and time perspectives also play an important role.

2. Process mining is not just a specific type of data mining.
Process mining can be seen as the "missing link" between data mining and traditional model-driven BPM. Most data mining techniques are not process-centric at all. Process models potentially exhibiting concurrency are incomparable to simple data mining structures such as decision trees and association rules. Therefore, completely new types of representations and algorithms are needed.

3. Process mining is not limited to offline analysis.
Process mining techniques extract knowledge from historical event data. Although "post mortem" data is used, the results can be applied to running cases. For example, the completion time of a partially handled customer order can be predicted using a discovered process model.

replenishment order, it may be interesting to know the supplier or the number of products ordered. The *time perspective* is concerned with the timing and frequency of events. When events bear timestamps it is possible to discover bottlenecks, measure service levels, monitor the utilization of resources, and predict the remaining processing time of running cases.

There are some common misconceptions related to process mining. Some vendors, analysts, and researchers limit the scope of process mining to a special data mining technique for process discovery that can only be used for offline analysis. This is *not* the case, therefore, we emphasize the three characteristics in the box on the previous page.

To position process mining, we use the Business Process Management (BPM) life-cycle shown in Fig. 4. The BPM life-cycle shows seven phases of a business process and its corresponding information system(s). In the *(re)design phase* a new process model is created or an existing process model is adapted. In the *analysis phase* a candidate model and its alternatives are analyzed. After the (re)design phase, the model is implemented (*implementation phase*) or an existing system is (re)configured (*reconfiguration phase*). In the *execution phase* the designed model is enacted. During the execution phase the process is *monitored*. Moreover, smaller adjustments may be made without redesigning the process (*adjustment phase*). In the *diagnosis phase* the enacted process is analyzed and the output of this phase may trigger a new process redesign phase. Process mining is a valuable tool for most of the phases shown in Fig. 4. Obviously, the diagnosis phase can benefit from process mining. However, process mining is not limited to the diagnosis phase. For example, in the execution phase, process mining techniques can be used for *operational support*. Predictions and recommendations based on models learned using historic information can be used to influence running cases. Similar forms of decision support can be used to adjust processes and to guide process (re)configuration.

Whereas Fig. 4 shows the overall BPM life-cycle, Fig. 5 focuses on the concrete process mining activities and artifacts. Figure 5 describes the possible stages in a process mining project. Any process mining project starts with a planning and a justification for this planning (Stage 0). After initiating the project, event data, models, objectives, and questions need to be extracted from systems, domain experts, and management (Stage 1). This requires an understanding of the available data ("What data can be used for analysis?") and an understanding of the domain ("What are the important questions?") and results in the artifacts shown in Fig. 5 (i.e., historical data, handmade models, objectives, and questions). In Stage 2 the control-flow model is constructed and linked to the event log. Here automated process discovery techniques can be used. The discovered process model may already provide answers to some of the questions and trigger redesign or adjustment actions.

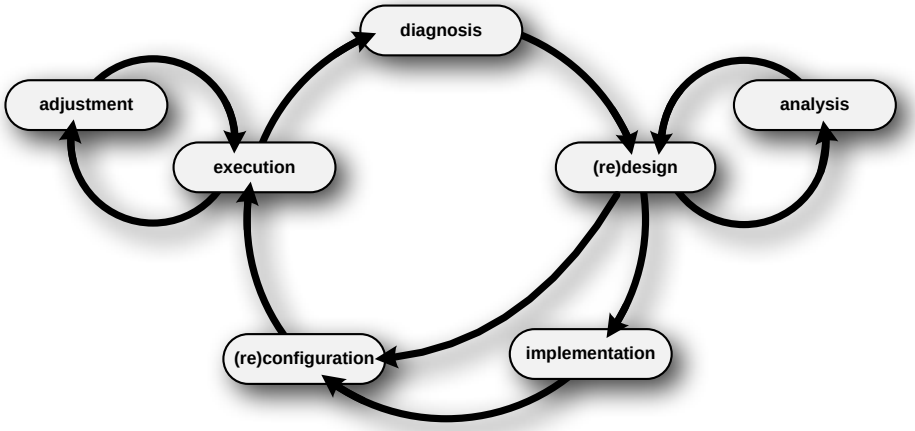


Figure 4: The BPM life-cycle identifying the various phases of a business process and its corresponding information system(s); process mining (potentially) plays a role in all phases (except for the implementation phase).

Guiding Principles:

GP1: Event Data Should be Treated as First-Class Citizens

GP2: Log Extraction Should Be Driven by Questions

GP3: Concurrency, Choice and Other Basic Control-Flow Constructs Should be Supported

GP4: Events Should Be Related to Model Elements

GP5: Models Should Be Treated as Purposeful Abstractions of Reality

GP6: Process Mining Should Be a Continuous Process

Moreover, the event log may be filtered or adapted using the model (e.g., removing rare activities or outlier cases, and inserting missing events). Sometimes significant efforts are needed to correlate events belonging to the same process instance. The remaining events are related to entities of the process model. When the process is relatively structured, the control-flow model may be extended with other perspectives (e.g., data, time, and resources) during Stage 3. The relation between the event log and the model established in Stage 2 is used to extend the model (e.g., timestamps of associated events are used to estimate waiting times for activities). This may be used to answer additional questions and may trigger additional actions. Ultimately, the models constructed in Stage 3 may be used for operational support (Stage 4). Knowledge extracted from historical event data is combined with information about running cases. This may be used to intervene, predict, and recommend. Stages 3 and 4 can only be reached if the process is sufficiently stable and structured.

Currently, there are techniques and tools that can support all stages shown in Fig. 5. However, process mining is a relatively new paradigm and most of the currently available tools are still rather immature. Moreover, prospective users are often not aware of the potential and the limitations of process

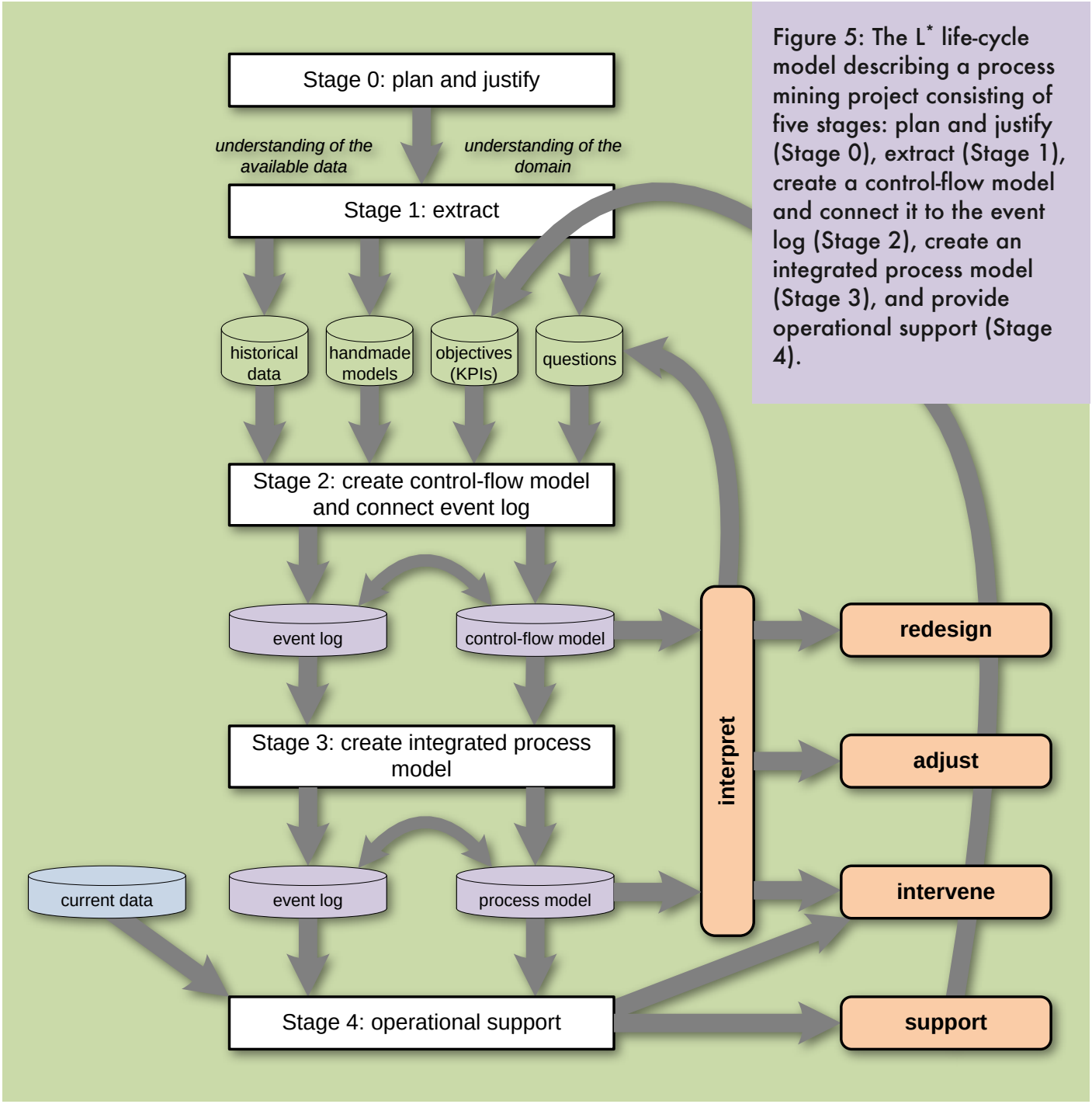


Figure 5: The L* life-cycle model describing a process mining project consisting of five stages: plan and justify (Stage 0), extract (Stage 1), create a control-flow model and connect it to the event log (Stage 2), create an integrated process model (Stage 3), and provide operational support (Stage 4).

mining. Therefore, this manifesto catalogs some guiding principles (cf. next section) and challenges (cf. page 10) for users of process mining techniques as well as researchers and developers that are interested in advancing the state-of-the-art.

3. Guiding Principles

As with any new technology, there are obvious mistakes that can be made when applying process mining in real-life settings. Therefore, we list six *guiding principles* to prevent users/ analysts from making such mistakes.

GP1: Event Data Should Be Treated as First-Class Citizens

Starting point for any process mining activity are the events recorded. We refer to collections of events as *event logs*, however, this does not imply that events need to be stored in dedicated log files. Events may be stored in database tables, message logs, mail archives, transaction logs, and other data sources. More important than the storage format, is the *quality* of such event logs. The quality of a process mining result heavily depends on the input. Therefore, event logs should be treated as *first-class citizens* in the information systems supporting the processes to be analyzed. Unfortunately, event logs are often merely a "by-product" used for debugging or profiling. For example, the medical devices of Philips Healthcare record events simply because software developers have inserted "print statements" in the code. Although there are some informal guidelines for adding such statements to the code, a more systematic approach is needed to improve the quality of event logs. Event data should be viewed as first-class citizens (rather than second-class citizens).

Level	Characterization	Examples
★★★★★	Highest level: the event log is of excellent quality (i.e., trustworthy and complete) and events are well-defined. Events are recorded in an automatic, systematic, reliable, and safe manner. Privacy and security considerations are addressed adequately. Moreover, the events recorded (and all of their attributes) have clear semantics. This implies the existence of one or more ontologies. Events and their attributes point to this ontology.	Semantically annotated logs of BPM systems.
★★★★	Events are recorded automatically and in a systematic and reliable manner, i.e., logs are trustworthy and complete. Unlike the systems operating at level ★★★, notions such as process instance (case) and activity are supported in an explicit manner.	Events logs of traditional BPM/ workflow systems.
★★★	Events are recorded automatically, but no systematic approach is followed to record events. However, unlike logs at level ★★, there is some level of guarantee that the events recorded match reality (i.e., the event log is trustworthy but not necessarily complete). Consider, for example, the events recorded by an ERP system. Although events need to be extracted from a variety of tables, the information can be assumed to be correct (e.g., it is safe to assume that a payment recorded by the ERP actually exists and vice versa).	Tables in ERP systems, event logs of CRM systems, transaction logs of messaging systems, event logs of high-tech systems, etc.
★★	Events are recorded automatically, i.e., as a by-product of some information system. Coverage varies, i.e., no systematic approach is followed to decide which events are recorded. Moreover, it is possible to bypass the information system. Hence, events may be missing or not recorded properly.	Event logs of document and product management systems, error logs of embedded systems, worksheets of service engineers, etc.
★	Lowest level: event logs are of poor quality. Recorded events may not correspond to reality and events may be missing. Event logs for which events are recorded by hand typically have such characteristics.	Trails left in paper documents routed through the organization ("yellow notes"), paper-based medical records, etc.

Table 1: Maturity levels for event logs.

There are several criteria to judge the quality of event data. Events should be *trustworthy*, i.e., it should be safe to assume that the recorded events actually happened and that the attributes of events are correct. Event logs should be *complete*, i.e., given a particular scope, no events may be missing. Any recorded event should have well-defined *semantics*. Moreover, the event data should be *safe* in the sense that privacy and security concerns are addressed when recording the events. For example, actors should be aware of the kind of events being recorded and the way they are used.

Table 1 defines five event log maturity levels ranging from excellent quality (★★★★★) to poor quality (★). For example, the event logs of Philips Healthcare reside at level ★★★, i.e., events are recorded automatically and the recorded behavior matches reality, but no systematic approach is used to assign semantics to events and to ensure coverage at a particular level. Process mining techniques can be applied to logs at levels ★★★★★, ★★★★, and ★★★. In principle, it is also possible to apply process mining using event logs at level ★★ or ★. However, the analysis of such logs is typically problematic and the results are not trustworthy. In fact, it does not make much sense to apply process mining to logs at level ★.

In order to benefit from process mining, organizations should aim at event logs at the highest possible quality level.

GP2: Log Extraction Should Be Driven by Questions

As shown in Fig. 5, process mining activities need to be driven by questions. Without concrete questions it is very difficult to extract meaningful event data. Consider, for example, the thousands of tables in the database of an ERP system like SAP. Without concrete questions it is impossible to select the tables relevant for data extraction.

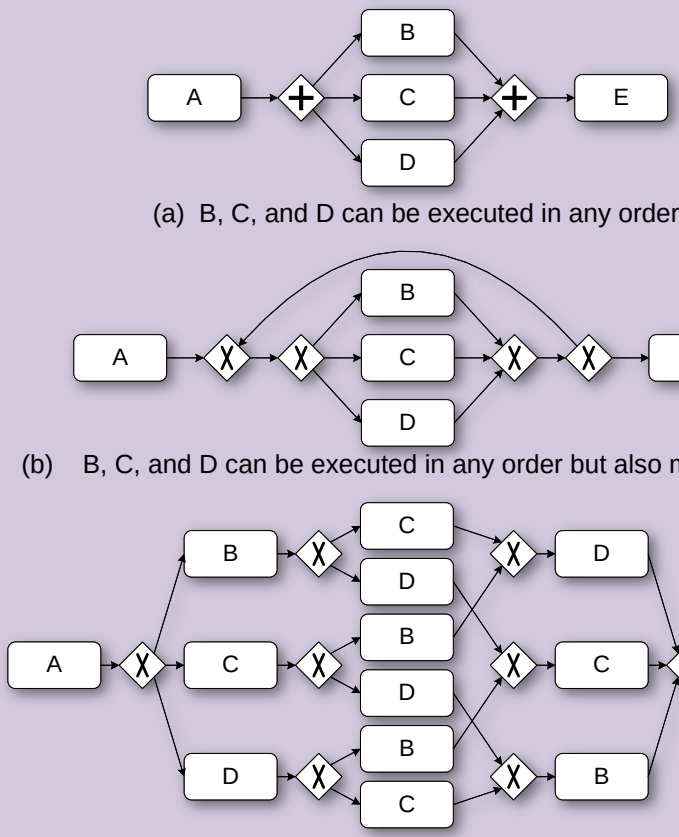
A process model such as the one shown in Fig. 1 describes the life-cycle of cases (i.e., process instances) of a particular type. Hence, before applying any process mining technique one needs to choose the type of cases to be analyzed. This choice should be driven by the questions that need to be

answered and this may be non-trivial. Consider, for example, the handling of customer orders. Each customer order may consist of multiple order lines as the customer may order multiple products in one order. One customer order may result in multiple deliveries. One delivery may refer to order lines of multiple orders. Hence, there is a many-to-many relationship between orders and deliveries and a one-to-many relationship between orders and order lines. Given a database with event data related to orders, order lines, and deliveries, there are different process models that can be discovered. One can extract data with the goal to describe the life-cycle of individual orders. However, it is also possible to extract data with the goal to discover the life-cycle of individual order lines or the life-cycle of individual deliveries.

GP3: Concurrency, Choice and Other Basic Control-Flow Constructs Should be Supported

A plethora of process modeling languages exists (e.g., BPMN, EPCs, Petri nets, BPEL, and UML activity diagrams). Some of these languages provide many modeling elements (e.g., BPMN offers more than 50 distinct graphical elements) whereas others are very basic (e.g., Petri nets are composed of only three different elements: places, transitions, and arcs). The control-flow description is the backbone of any process model. Basic workflow constructs (also known as *patterns*) supported by all mainstream languages are sequence, parallel routing (AND-splits/joins), choice (XOR-splits/joins), and loops. Obviously, these patterns should be supported by process mining techniques. However, some techniques are not able to deal with concurrency and support only Markov chains/transition systems.

Figure 6 shows the effect of using process mining techniques unable to discover concurrency (no AND-split/joins). Consider an event log L={ ⟨A, B, C, D, E⟩ , ⟨A, B, D, C, E⟩ , ⟨A, C, B, D, E⟩ , ⟨A, C, D, B, E⟩ , ⟨A, D, B, C, E⟩ , ⟨A, D, C, B, E⟩ }. L contains cases that start with A and end with E. Activities B, C, and D occur in any order in-between A and E. The BPMN model in Fig. 6(a) shows a compact representation of the underlying process using two AND gateways. Suppose that the process mining technique does not support AND gateways. In this case, the other two BPMN models in Fig. 6 are obvious candidates. The BPMN model in Fig. 6(b) is compact but allows for too much behavior (e.g., cases such as ⟨A, B, B, B, E⟩ are possible according to the model but are not likely according to the event log). The BPMN model in Fig. 6(c) allows for the cases in L, but encodes all sequences explicitly, so it is not a compact representation of the log. The example shows that for real-life models having dozens of potentially concurrent activities the resulting models are severely underfitting (i.e., allow for too much behavior) and/or extremely complex if concurrency is not supported.



(a) B, C, and D can be executed in any order

(b) B, C, and D can be executed in any order but also multiple times

(c) B, C, and D can be executed in any order, but activities need to be duplicated to model all observed sequences.

Figure 6: Example illustrating problems when concurrency (i.e., AND-splits/joins) cannot be expressed directly. In the example just three activities (B, C, and D) are concurrent. Imagine the resulting process models when there are 10 concurrent activities (2^{10}=1024 states and 10! = 3,628,800 possible execution sequences).

As is illustrated by Fig. 6, it is important to support at least the basic workflow patterns. Besides the basic patterns mentioned it is also desirable to support OR-splits/joins, because these provide a compact representation of inclusive decisions and partial synchronizations.

GP4: Events Should Be Related to Model Elements

As indicated in Sect. 2, it is a misconception that process mining is limited to control-flow discovery. As shown in Fig. 1, the discovered process model may cover various perspectives (organizational perspective, time perspective, data perspective, etc.). Moreover, discovery is just one of the three types of process mining shown in Fig. 3. The other two types of process mining (conformance checking and enhancement) heavily rely on the relationship between *elements in the*

model and *events in the log*. This relationship may be used to "replay" the event log on the model. Replay may be used to reveal discrepancies between an event log and a model, e.g., some events in the log are not possible according to the model. Techniques for conformance checking quantify and diagnose such discrepancies. Timestamps in the event log can be used to analyze the temporal behavior during replay. Time differences between causally related activities can be used to add expected waiting times to the model. These examples show that the relation between events in the log and elements in the model serves as a starting point for different types of analysis.

In some cases it may be non-trivial to establish such a relationship. For example, an event may refer to two different activities or it is unclear to which activity it refers. Such ambiguities need to be removed in order to interpret process mining results properly. Besides the problem of relating events to activities, there is the problem of relating events to process instances. This is commonly referred to as *event correlation*.

GP5: Models Should Be Treated as Purposeful Abstractions of Reality

A model derived from event data provides a *view on reality*. Such a view should provide a purposeful abstraction of the behavior captured in the event log. Given an event log, there may be multiple views that are useful. Moreover, the various stakeholders may require different views. In fact, models discovered from event logs should be seen as "maps" (like geographic maps). This guiding principle provides important insights, two of which are described in the remainder.

First of all, it is important to note that there is no such thing as "the map" for a particular geographic area. Depending on the intended use there are different maps: road maps, hiking maps, cycling maps, etc. All of these maps show a view on the same reality and it would be absurd to assume that there would be such a thing as "the perfect map". The same holds for process models: the model should emphasize the things relevant for a particular type of user. Discovered models may focus on different perspectives (control-flow, data flow, time, resources, costs, etc.) and show these at different levels of granularity and precision, e.g., a manager may want to see a coarse informal process model focusing on costs whereas a process analyst may want to see a detailed process model focusing on deviations from the normal flow. Also note that different stakeholders may want to view a process at different levels: *strategic level* (decisions at this level have long-term effects and are based on aggregate event data over a longer period), *tactical level* (decisions at this level have medium-term effects and are mostly based on recent data), and *operational level* (decisions at this level have immediate effects and are based on event data related to running cases).

Second, it is useful to adopt ideas from cartography when it comes to producing understandable maps. For example, road maps abstract from less significant roads and cities. Less significant things are either left out or dynamically clustered into aggregate shapes (e.g., streets and suburbs amalgamate into cities). Cartographers not only eliminate irrelevant details, but also use colors to highlight important features. Moreover, graphical elements have a particular size to indicate their significance (e.g., the sizes of lines and dots may vary). Geographical maps also have a clear interpretation of the x-axis and y-axis, i.e., the layout of a map is not arbitrary as the coordinates of elements have a meaning. All of this is in stark contrast with mainstream process models which are typically not using color, size, and location features to make models more understandable. However, ideas from cartography can easily be incorporated in the construction of discovered process maps. For example, the size of an activity can be used to reflect its frequency or some other property indicating its significance (e.g., costs or resource use). The width of an arc can reflect the importance of the corresponding causal dependency, and the coloring of arcs can be used to highlight bottlenecks.

The above observations show that it is important to select the right representation and fine-tune it for the intended audience. This is important for visualizing results to end users and for guiding discovery algorithms towards suitable models (see also Challenge C5).

GP6: Process Mining Should Be a Continuous Process

Process mining can help to provide meaningful "maps" that are directly connected to event data. Both historical event data and current data can be projected onto such models. Moreover, processes change while they are being analyzed. Given the dynamic nature of processes, it is not advisable to see process mining as a one-time activity. The goal should not be to create a fixed model, but to breathe life into process models so that users and analysts are encouraged to look at them on a daily basis.

Compare this to the use of mashups using geo-tagging. There are thousands of mashups using Google Maps (e.g., applications projecting information about traffic conditions, real estate, fastfood restaurants, or movie showtimes onto a selected map). People can seamlessly zoom in and out using such maps and interact with them (e.g., traffic jams are projected onto the map and the user can select a particular problem to see details). It should also be possible to conduct process mining based on real-time event data. Using the "map metaphor", we can think of events having GPS coordinates that can be projected on maps in real time. Analogous to car navigation systems, process mining tools can help end users (a) by navigating through processes, (b) by projecting dynamic information onto process maps (e.g., showing "traffic jams" in business processes), and (c) by providing predictions regarding running cases (e.g., estimating the "arrival time" of a case that is delayed). These examples demonstrate that it is a pity to not use process models more actively. Therefore, process mining should be viewed as a continuous process providing actionable information according to various time scales (minutes, hours, days, weeks, and months).

4. Challenges

Process mining is an important tool for modern organizations that need to manage non-trivial operational processes. On the one hand, there is an incredible growth of event data. On the other hand, processes and information need to be aligned perfectly in order to meet requirements related to compliance, efficiency, and customer service. Despite the applicability of process mining there are still important challenges that need to be addressed; these illustrate that process mining is an emerging discipline. In the remainder, we list some of these challenges. This list is not intended to be complete and, over time, new challenges may emerge or existing challenges may disappear due to advances in process mining.

C1: Finding, Merging, and Cleaning Event Data

It still takes considerable efforts to extract event data suitable for process mining. Typically, several hurdles need to be overcome:

- Data may be *distributed* over a variety of sources. This information needs to be merged. This tends to be problematic when different identifiers are used in the different data sources. For example, one system uses name and birthdate to identify a person whereas another system uses the person's social security number.
- Event data are often "object centric" rather than "process centric". For example, individual products, pallets, and containers may have RFID tags and recorded events refer to these tags. However, to monitor a particular customer order such object-centric events need to be merged and preprocessed.
- Event data may be *incomplete*. A common problem is that events do not explicitly point to process instances. Often it is possible to derive this information, but this may take considerable efforts. Also time information may be missing for some events. One may need to interpolate timestamps in order to still use the timing information available.
- An event log may contain *outliers*, i.e., exceptional behavior also referred to as *noise*. How to define outliers? How to detect such outliers? These questions need to be answered to clean event data.
- Logs may contain events at *different levels of granularity*. In the event log of a hospital information system events may refer to simple blood tests or to complex surgical procedures. Also timestamps may have different levels of granularity ranging from milliseconds precision (28-9-2011:h11m28s32ms342) to coarse date information (28-9-2011).
- Events occur in a particular *context* (weather, workload, day of the week, etc.). This context may explain certain phenomena, e.g., the response time is longer than usual because of work-in-progress or holidays. For analysis, it is desirable to incorporate this context. This implies the merging of event data with contextual data. Here the "curse of dimensionality" kicks in as analysis becomes intractable when adding too many variables.

Better tools and methodologies are needed to address the above problems. Moreover, as indicated earlier, organizations need to treat event logs as first-class citizens rather than some by-product. The goal is to obtain ★★★★★ event logs (see Table 1). Here, the lessons learned in the context of datawarehousing are useful to ensure high-quality event logs. For example, simple checks during data entry can help to reduce the proportion of incorrect event data significantly.

C2: Dealing with Complex Event Logs Having Diverse Characteristics

Event logs may have very different characteristics. Some event logs may be extremely large making it difficult to handle them whereas other event logs are so small that not enough data is available to make reliable conclusions.

In some domains, mind-boggling quantities of events are recorded. Therefore, additional efforts are needed to improve performance and scalability. For example, ASML is continuously monitoring all of its wafer scanners. These wafer scanners are used by various organizations (e.g., Samsung and Texas Instruments) to produce chips (approx. 70% of chips are produced using ASML's wafer scanners). Existing tools have difficulties dealing with the petabytes of data collected in such domains. Besides the number of events recorded there are other characteristics such as the average number of events per case, similarity among cases, the number of unique events, and the number of unique paths. Consider an event log L1 with the following characteristics: 1000 cases, on average 10 events per case, and little variation (e.g., several cases follow the same or very similar paths). Event log L2 contains just 100 cases, but on average there are 100 events per case and all cases follow a unique path. Clearly, L2 is much more difficult to analyze than L1 even though the two logs have similar sizes (approximately 10,000 events).

As event logs contain only sample behavior, they should not be assumed to be complete. Process mining techniques need to deal with incompleteness by using an "open world assumption": the fact that something did not happen does not mean that it cannot happen. This makes

Challenges:

C1: Finding, Merging, and Cleaning Event Data

C2: Dealing with Complex Event Logs Having Diverse Characteristics

C3: Creating Representative Benchmarks

C4: Dealing with Concept Drift

C5: Improving the Representational Bias Used for Process Discovery

C6: Balancing Between Quality Criteria such as Fitness, Simplicity, Precision, and Generalization

C7: Cross-Organizational Mining

C8: Providing Operational Support

C9: Combining Process Mining With Other Types of Analysis

C10: Improving Usability for Non-Experts

C11: Improving Understandability for Non-Experts

it challenging to deal with small event logs with a lot of variability.

As mentioned before, some logs contain events at a very low abstraction level. These logs tend to be extremely large and the individual low-level events are of little interest to the stakeholders. Therefore, one would like to aggregate low-level events into high-level events. For example, when analyzing the diagnostic and treatment processes of a particular group of patients one may not be interested in the individual tests recorded in the information system of the hospital's laboratory.

At this point in time, organizations need to use a trial-and-error approach to see whether an event log is suitable for process mining. Therefore, tools should allow for a quick feasibility test given a particular data set. Such a test should indicate potential performance problems and warn for logs that are far from complete or too detailed.

C3: Creating Representative Benchmarks

Process mining is an emerging technology. This explains why good benchmarks are still missing. For example, dozens of process discovery techniques are available and different vendors offer different products, but there is no consensus on the quality of these techniques. Although there are huge differences in functionality and performance, it is difficult to compare the different techniques and tools. Therefore, good benchmarks consisting of example data sets and representative quality criteria need to be developed.

For classical data mining techniques, many good benchmarks are available. These benchmarks have stimulated tool providers and researchers to improve the performance of their techniques. In the case of process mining this is more challenging. For example, the relational model introduced by Codd in 1969 is simple and widely supported. As a result it takes little effort to convert data from one database to another and there are no interpretation problems. For processes such a simple model is missing. Standards proposed for process modeling are much more complicated and few vendors support exactly the same set of concepts. Processes are simply more complex than tabular data.

Nevertheless, it is important to create representative benchmarks for process mining. Some initial work is already available. For example, there are various metrics for measuring the quality of process mining results (fitness, simplicity, precision, and generalization). Moreover, several event logs are publicly available (cf. www.processmining.org). See for example the event log used for the first Business Process Intelligence Challenge (BPIC'11) organized by the task force (cf. doi:10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54).

On the one hand, there should be benchmarks based on real-life data sets. On the other hand, there is the need to create synthetic datasets capturing particular characteristics. Such synthetic datasets help to develop process mining techniques that are tailored towards incomplete event logs, noisy event logs, or specific populations of processes.

Besides the creation of representative benchmarks, there also needs to be more consensus on the criteria used to judge the quality of process mining results (also see Challenge C6). Moreover, *cross-validation* techniques from data mining can be adapted to judge the result. Consider for example *k*-fold checking. One can split the event log in *k* parts. $k-1$ parts can be used to learn a process model and conformance checking techniques can be used to judge the result with respect to the remaining part. This can be repeated *k* times, thus providing some insights into the quality of the model.

C4: Dealing with Concept Drift

The term *concept drift* refers to the situation in which the process is changing while being analyzed. For instance, in the beginning of the event log two activities may be concurrent whereas later in the log these activities become sequential. Processes may change due to periodic/seasonal changes (e.g., "in December there is more demand" or "on Friday afternoon there are fewer employees available") or due to changing conditions (e.g., "the market is getting more competitive"). Such changes impact processes and it is vital to detect and analyze them. Concept drift in a process can be discovered by splitting the event log into smaller logs and analyzing the "footprints" of the smaller logs. Such "second order" analysis requires much more event data. Nevertheless, few processes are in steady state and understanding concept drift is of prime importance for the management of processes. Therefore, additional research and tool support are needed to adequately analyze concept drift.

C5: Improving the Representational Bias Used for Process Discovery

A process discovery technique produces a model using a particular language (e.g., BPMN or Petri nets). However, it is important to separate the visualization of the result from the representation used during the actual discovery process. The selection of a target language often encompasses several implicit assumptions. It limits the search space; processes that cannot be represented by the target language cannot be discovered. This so-called "representational bias" used during the discovery process should be a conscious choice and should not be (only) driven by the preferred graphical representation.

Consider for example Fig. 6: whether the target language allows for concurrency or not may have an effect on both the visualization of the discovered model and the class of models considered by the algorithm. If the representational bias does not allow for concurrency (Fig. 6(a) is not possible) and does not allow for multiple activities having the same label (Fig. 6(c) is not possible), then only problematic models such as the one shown in Fig. 6(b) are possible. This example shows that a more careful and refined selection of the representational bias is needed.

C6: Balancing Between Quality Criteria such as Fitness, Simplicity, Precision, and Generalization

Event logs are often far from being complete, i.e., only example behavior is given. Process models typically allow for an exponential or even infinite number of different traces (in case of loops). Moreover, some traces may have a much lower probability than others. Therefore, it is unrealistic to assume that every possible trace is present in the event log. To illustrate that it is impractical to take complete logs for granted, consider a process consisting of 10 activities that can be executed in parallel and a corresponding log that contains information about 10,000 cases. The total number of possible interleavings in the model with 10 concurrent activities is 10! = 3,628,800. Hence, it is impossible that each interleaving is present in the log as there are fewer cases (10,000) than potential traces (3,628,800). Even if there are millions of cases in the log, it is extremely unlikely that all possible variations are present. An additional complication is that some alternatives are less frequent than others. These may be considered as "noise". It is impossible to build a reasonable model for such noisy behaviors. The discovered model needs to abstract from this; it is better to investigate low frequency behavior using conformance checking.

Noise and incompleteness make process discovery a challenging problem. In fact, there are four competing quality dimensions: (a) fitness, (b) simplicity, (c) precision, and (d) generalization. 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. The *simplest* model that can explain the behavior seen in the log is the best model. This principle is known as Occam's Razor. Fitness and simplicity alone are not sufficient to judge the quality of a discovered process model. For example, it is very easy to construct an extremely simple Petri net ("flower model") that is able to replay all traces in an event log (but also any other event log referring to the same set of activities). Similarly, it is undesirable to have a model that only allows for the exact behavior seen in the event log. Remember that the log contains only example behavior and that many traces that are possible may not have been seen yet. A model is *precise* if it does not allow for "too much" behavior. Clearly, the "flower model" lacks precision. A model that is not precise is "underfitting". Underfitting is the problem that the model over-generalizes the example behavior in the log (i.e., the model allows for behaviors very different from what was seen in the log). A model should generalize and not restrict behavior to just the examples seen in the log. A model that does not *generalize* is "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).

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.

C7: Cross-Organizational Mining

Traditionally, process mining is applied within a single organization. However, as service technology, supply-chain integration, and cloud computing become more widespread, there are scenarios where the event logs of multiple organizations are available for analysis. In principle, there are two settings for *cross-organizational process mining*.

First of all, we may consider the collaborative setting where different organizations work together to handle process instances. One can think of such a cross-organizational process as a "jigsaw puzzle", i.e., the overall process is cut into parts and distributed over organizations that need to cooperate to successfully complete cases. Analyzing the event log within one of these organizations involved is insufficient. To discover end-to-end processes, the event logs of different organizations need to be merged. This is a non-trivial task as events need to be correlated across organizational boundaries.

Second, we may also consider the setting where different organizations are essentially executing the same process while sharing experiences, knowledge, or a common infrastructure. Consider for example Salesforce.com. The sales processes of many organizations are managed and supported by Salesforce. On the one hand, these organizations share an infrastructure (processes, databases, etc.). On the other hand, they are not forced to follow a strict process model as the system can be configured to support variants of the same process. As another example, consider the basic processes executed within any municipality (e.g., issuing building permits). Although all municipalities in a country need to support the same basic set of processes, there may be also be differences. Obviously, it is interesting to analyze such variations among different organizations. These organizations can learn from one another and service providers may improve their services and offer value-added services based on the results of cross-organizational process mining.

New analysis techniques need to be developed for both types of cross-organizational process mining. These techniques should also consider privacy and security issues. Organizations may not want to share information for competitive reasons or due to a lack of trust. Therefore, it is important to develop privacy-preserving process mining techniques.

C8: Providing Operational Support

Initially, the focus of process mining was on the analysis of historical data. Today, however, many data sources are updated in (near) real-time and sufficient computing power is available to analyze events when they occur. Therefore, process mining should not be restricted to off-line analysis and can also be used for online operational support. Three operational support activities can be identified: *detect*, *predict*, and *recommend*. The moment a case deviates from the predefined process, this can be detected and the system can generate an alert. Often one would like to generate such

notifications immediately (to still be able to influence things) and not in an off-line fashion. Historical data can be used to build predictive models. These can be used to guide running process instances. For example, it is possible to predict the remaining processing time of a case. Based on such predictions, one can also build recommender systems that propose particular actions to reduce costs or shorten the flow time. Applying process mining techniques in such an online setting creates additional challenges in terms of computing power and data quality.

C9: Combining Process Mining With Other Types of Analysis

Operations management, and in particular operations research, is a branch of management science heavily relying on modeling. Here a variety of mathematical models ranging from linear programming and project planning to queueing models, Markov chains, and simulation are used. Data mining can be defined as "the analysis of (often large) data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner". A wide variety of techniques have been developed: classification (e.g., decision tree learning), regression, clustering (e.g., k-means clustering) and pattern discovery (e.g., association rule learning).

Both fields (operations management and data mining) provide valuable analysis techniques. The challenge is to combine the techniques in these fields with process mining. Consider for example simulation. Process mining techniques can be used to learn a simulation model based on historical data. Subsequently, the simulation model can be used to provide operational support. Because of the close connection between event log and model, the model can be used to replay history and one can start simulations from the current state thus providing a "fast forward button" into the future based on live data.

Similarly, it is desirable to combine process mining with *visual analytics*. Visual analytics combines automated analysis with interactive visualizations for a better understanding of large and complex data sets. Visual analytics exploits the amazing capabilities of humans to see patterns in unstructured data. By combining automated process mining techniques with interactive visual analytics, it is possible to extract more insights from event data.

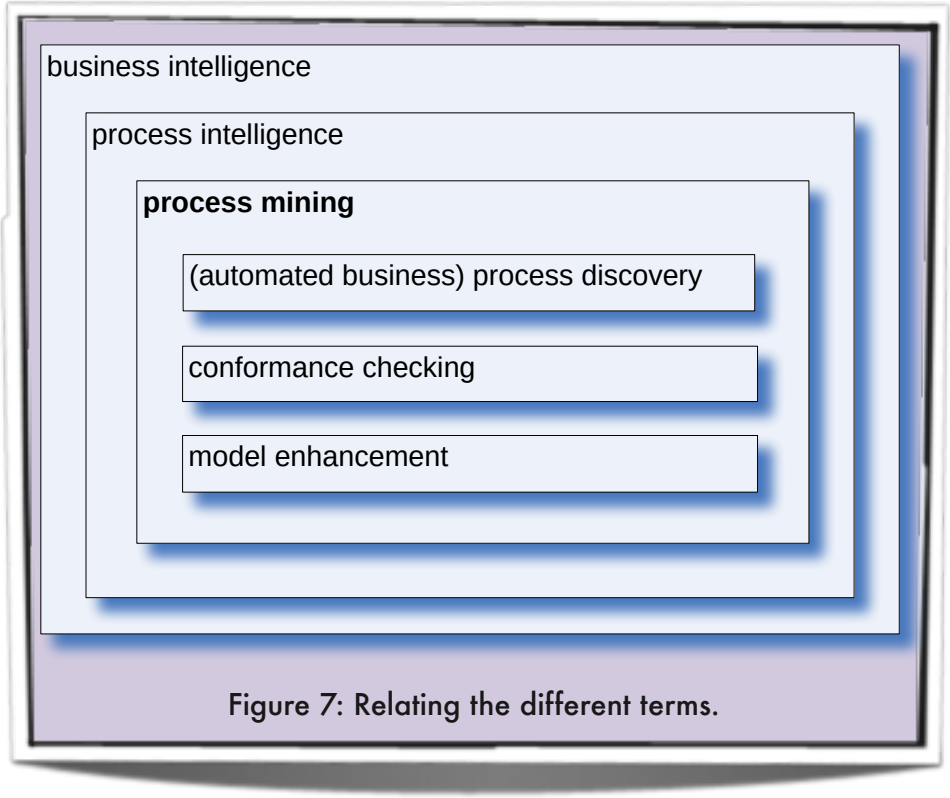


Figure 7: Relating the different terms.

C10: Improving Usability for Non-Experts

One of the goals of process mining is to create "living process models", i.e., process models that are used on a daily basis rather than static models that end up in some archive. New event data can be used to discover emerging behavior. The link between event data and process models allows for the projection of the current state and recent activities onto up-to-date models. Hence, end-users can interact with the results of process mining on a day-to-day basis. Such interactions are very valuable, but also require intuitive user interfaces. The challenge is to hide the sophisticated process mining algorithms behind user-friendly interfaces that automatically set parameters and suggest suitable types of analysis.

C11: Improving Understandability for Non-Experts

Even if it is easy to generate process mining results, this does not mean that the results are actually useful. The user may have problems understanding the output or is tempted to infer incorrect conclusions. To avoid such problems, the results should be presented using a suitable representation (see also GP5). Moreover, the trustworthiness of the results should always be clearly indicated. There may be too little data to justify particular conclusions. In fact, existing process discovery techniques typically do not warn for a low fitness or for overfitting. They always show a model, even when it is clear that there is too little data to justify any conclusions.

Epilogue

The IEEE Task Force on Process Mining aims to (a) promote the application of process mining, (b) guide software developers, consultants, business managers, and end-users when using state-of-the-art techniques, and (c) stimulate research on process mining. This manifesto states the main principles and intentions of the task force. After introducing the topic of process mining, the manifesto catalogs some guiding principles (Section 3) and challenges (Section 4). The guiding principles can be used in order to avoid obvious mistakes. The list of challenges is intended to direct research and development efforts. Both aim to increase the maturity level of process mining.

To conclude, a few words on terminology. The following terms are used in the process mining space: workflow mining, (business) process mining, automated (business) process

discovery, and (business) process intelligence. Different organizations seem to use different terms for overlapping concepts. For example, Gartner is promoting the term "Automated Business Process Discovery" (ABPD) and Software AG is using "Process Intelligence" to refer to their controlling platform. The term "workflow mining" seems less suitable as the creation of workflow models is just one of the many possible applications of process mining. Similarly, the addition of the term "business" narrows the scope to certain applications of process mining. There are numerous applications of process mining (e.g., analyzing the use of high-tech systems or analyzing websites) where this addition seems to be inappropriate. Although process discovery is an important part of the process mining spectrum, it is only one of the many use cases. Conformance checking, prediction, organizational mining, social network analysis, etc. are other use cases that extend beyond process discovery.

Figure 7 relates some of the terms just mentioned. All technologies and methods that aim at providing actionable information that can be used to support decision making can be positioned under the umbrella of Business Intelligence (BI). (Business) process intelligence can be seen as the combination of BI and BPM, i.e., BI techniques are used to analyze and improve processes and their management. Process mining can be seen as a concretization of process intelligence taking event logs as a starting point. (Automated business) process discovery is just one of the three basic types of process mining. Figure 7 may be a bit misleading in the sense that most BI tools do not provide process mining functionality as described in this document. The term BI is often conveniently skewed towards a particular tool or method covering only a small part of the broader BI spectrum.

There may be commercial reasons for using alternative terms. Some vendors may also want to emphasize a particular aspect (e.g., discovery or intelligence). However, to avoid confusion, it is better to use the term "process mining" for the discipline covered by this manifesto.

Glossary

Activity: a well-defined step in the process. Events may refer to the start, completion, cancelation, etc. of an activity for a specific process instance.
Automated Business Process Discovery: see Process Discovery.
Business Intelligence (BI): broad collection of tools and methods that use data to support decision making.
Business Process Intelligence: see Process Intelligence.
Business Process Management (BPM): the discipline that combines knowledge from information technology and knowledge from management sciences and applies both to operational business processes.
Case: see Process Instance.
Concept Drift: the phenomenon that processes often change over time. The observed process may gradually (or suddenly) change due to seasonal changes or increased competition, thus complicating analysis.
Conformance Checking: analyzing whether reality, as recorded in a log, conforms to the model and vice versa. The goal is to detect discrepancies and to measure their severity. Conformance checking is one of the three basic types of process mining.
Cross-Organizational Process Mining: the application of process mining techniques to event logs originating from different organizations.
Data Mining: the analysis of (often large) data sets to find unexpected relationships and to summarize the data in ways that provide new insights.
Event: an action recorded in the log, e.g., the start, completion, or cancelation of an activity for a particular process instance.
Event Log: collection of events used as input for process mining. Events do not need to be stored in a separate log file (e.g., events may be scattered over different database tables).
Fitness: a measure determining how well a given model allows for 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.
Generalization: a measure determining how well the model is able to allow for unseen behavior. An "overfitting" model is not able to generalize enough.

This Manifesto was originally published in "Business Process Management Workshops 2011, Lecture Notes in Business Information Processing, Vol. 99, Springer-Verlag, 2011, and has been translated into various languages. See the home page of IEEE Task Force on Process Mining: http://www.win.tue.nl/ieeetfpm/ for more information.

Model Enhancement: one of the three basic types of process mining. A process model is extended or improved using information extracted from some log. For example, bottlenecks can be identified by replaying an event log on a process model while examining the timestamps.
MXML: an XML-based format for exchanging event logs. XES replaces MXML as the new tool-independent process mining format.
Operational Support: on-line analysis of event data with the aim to monitor and influence running process instances. Three operational support activities can be identified: *detect* (generate an alert if the observed behavior deviates from the modeled behavior), *predict* (predict future behavior based on past behavior, e.g., predict the remaining processing time), and *recommend* (suggest appropriate actions to realize a particular goal, e.g., to minimize costs).
Precision: measure determining whether the model prohibits behavior very different from the behavior seen in the event log. A model with low precision is "underfitting".
Process Discovery: one of the three basic types of process mining. Based on an event log a process model is learned. For example, the α algorithm is able to discover a Petri net by identifying process patterns in collections of events.
Process Instance: the entity being handled by the process that is analyzed. Events refer to process instances. Examples of process instances are customer orders, insurance claims, loan applications, etc.
Process Intelligence: a branch of Business Intelligence focusing on Business Process Management.

Process Mining: techniques, tools, and methods to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs commonly available in today's (information) systems.
Representational Bias: the selected target language for presenting and constructing process mining results.
Simplicity: a measure operationalizing Occam's Razor, i.e., the simplest model that can explain the behavior seen in the log, is the best model. Simplicity can be quantified in various ways, e.g., number of nodes and arcs in the model.
XES: is an XML-based standard for event logs. The standard has been adopted by the IEEE Task Force on Process Mining as the default interchange format for event logs (cf. www.xes-standard.org).

Authors

Wil van der Aalst
Arya Adriansyah
Ana Karla Alves de Medeiros
Franco Arcieri
Thomas Baier
Tobias Blickle
Jagadeesh Chandra Bose
Peter van den Brand
Ronald Brandtjen
Joos Buijs
Andrea Burattin
Josep Carmona
Malu Castellanos
Jan Claes
Jonathan Cook
Nicola Costantini
Francisco Curbera
Ernesto Damiani
Massimiliano de Leoni
Pavlos Delias
Boudewijn van Dongen
Marlon Dumas
Schahram Dustdar
Dirk Fahland
Diogo R. Ferreira
Walid Gaaloul
Frank van Geffen
Sukriti Goel
Christian Günther
Antonella Guzzo
Paul Harmon
Arthur ter Hofstede
John Hoogland
Jon Espen Ingvaldsen
Koki Kato
Rudolf Kuhn
Akhil Kumar
Marcello La Rosa
Fabrizio Maggi
Donato Malerba
Ronny Mans
Alberto Manuel
Martin McCreesh
Paola Mello
Jan Mendling
Marco Montali
Hamid Motahari Nezhad
Michael zur Muehlen
Jorge Munoz-Gama
Luigi Pontieri
Joel Ribeiro
Anne Rozinat
Hugo Seguel Pérez
Ricardo Seguel Pérez
Marcos Sepúlveda
Jim Sinur
Pnina Soffer
Minseok Song
Alessandro Sperduti
Giovanni Stilo
Casper Stoel
Keith Swenson
Maurizio Talamo
Wei Tan
Chris Turner
Jan Vanthienen
George Varvaressos
Eric Verbeek
Marc Verdonk
Roberto Vigo
Jianmin Wang
Barbara Weber
Matthias Weidlich
Ton Weijters
Lijie Wen
Michael Westergaard
Moe Wynn