Chapter 11
Process Mining Software

The successful application of process mining relies on good tool support. Traditional Business Intelligence (BI) tools are data-centric and focus on rather simplistic forms of analysis. Mainstream data mining and machine learning tools provide more sophisticated forms of analysis, but are also not tailored towards the analysis and improvement of processes. Fortunately, there are dedicated process mining tools able to transform event data into actionable process-related insights. For example, ProM is an open-source process mining tool supporting all of the techniques mentioned in this book. Process discovery, conformance checking, social network analysis, organizational mining, clustering, decision mining, prediction, and recommendation are all supported by ProM plug-ins. However, the usability of the hundreds of available plug-ins varies and the complexity of the tool may be overwhelming for end-users. In recent years, several vendors released dedicated process mining tools (e.g., Celonis, Disco, EDS, Fujitsu, Minit, myInvenio, Perceptive, PPM, QPR, Rialto, and SNP). These tools typically provide less functionality than ProM, but are easier to use while focusing on data extraction, performance analysis and scalability. This chapter provides an overview of available tools and trends.

11.1 Process Mining Not Included!

This book revolves around the analysis of behavior based on event data. Fueled by the growing availability of data ("Big Data"), data science emerged as a new discipline. As discussed in Sect. 1.3, data science approaches tend to be process agnostic. Process mining aims for duality (yin and yang) between data-driven forces and process-centric forces (see Fig. 2.1). The process mining spectrum is broad and, as shown in the previous chapters, extends far beyond process discovery and conformance checking. Process mining connects data science and process science (see Fig. 1.7). Hence, it is inevitable that process mining objectives are overlapping with those of other approaches, methodologies, principles, methods, tools, and paradigms. In Sect. 2.5, we discussed the relation to BPM, BPR, BI, Big Data, data

W. van der Aalst, *Process Mining*, DOI 10.1007/978-3-662-49851-4_11

mining, Lean Six Sigma, etc. We posed questions like: "How does process mining compare to data mining?" (Sect. 2.5.2) and "How does process mining compare to Business Intelligence?" (Sect. 2.5.5). Books on data mining and BI seldom cover process mining techniques. The same holds for data mining and BI software. *Defining process mining as a particular type of machine learning, data mining or BI technique, will not extend the actual capabilities of (machine learning, data mining or BI) tools.* Software packages for machine learning and data mining *cannot* deal with process models (i.e., BPMN, EPC, UML, Petri nets, etc.) and do *not* support tasks like conformance checking. One needs dedicated process mining software for this: It is not included!

In the remainder of this chapter, we describe the capabilities of ProM and various commercial process mining tools. However, before doing so, we briefly discuss the market for BI products.

Forrester defines *Business Intelligence* (BI) in two ways. The broad definition provided by Forrester is "BI is a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making" [55]. Forrester also provides a second, more narrow, definition: "BI is a set of methodologies, processes, architectures, and technologies that leverage the output of information management processes for analysis, reporting, performance management, and information delivery" [55].

Some of the most widely used BI products are [56]: *IBM Cognos Business Intelligence* (IBM), *Oracle Business Intelligence* (Oracle), *SAP BusinessObjects* (SAP), *MS SQL Server/Power BI* (Microsoft), *MicroStrategy* (MicroStrategy), *QlikView* (QlikTech), *SAS Business Intelligence* (SAS), *TIBCO Spotfire Analytics* (TIBCO), *Jaspersoft BI Enterprise* (Jaspersoft), and *Pentaho BI Platform* (Pentaho). The typical functionality provided by these products includes:

- *ETL* (Extract, Transform, and Load). All products support the extraction of data from various sources. The extracted data is then transformed into a standard data format (typically a multidimensional table) and loaded into the BI system.
- *Ad-hoc querying.* Users can explore the data in an ad-hoc manner (e.g., drilling down and "slicing and dicing").
- *Reporting.* All BI products allow for the definition of standard reports. Users without any knowledge of the underlying data structures can simply generate such predefined reports. A report may contain various tables, graphs, and scorecards.
- *Interactive dashboards.* All BI products allow for the definition of dashboards consisting of tabular data and a variety of graphs. These dashboards are interactive, e.g., the user can change, refine, aggregate, and filter the current view using predefined controls.
- *Alert generation.* It is possible to define events and conditions that need to trigger an alert, e.g., when sales drop below a predefined threshold an e-mail is sent to the sales manager.

The mainstream BI products from vendors such as IBM, Oracle, SAP, and Microsoft do *not* support process mining. All of the systems mentioned earlier are *data-centric* and are *unaware* of the processes the data refers to. The focus is on fancy-looking dashboards and rather simple reports, instead of a deeper analysis of the data collected. This is surprising as the “I” in BI refers to “intelligence”. Unfortunately, the business ~~un~~intelligence market is dominated by large vendors that focus on monitoring and reporting rather than analytics. Data mining or statistical analysis are often added as an afterthought.

Most BI tools provide interfaces to data mining tools. For example, open-source BI products from organizations like Jaspersoft and Pentaho can connect to open-source data mining tools such as *WEKA* (Waikato Environment for Knowledge Analysis, weka.wikispaces.com), *RapidMiner* (www.rapidminer.com), *KNIME* (Konstanz Information Miner, www.knime.org), and *R* (www.r-project.org). These provide more “intelligence” than mainstream BI tools.

WEKA is a widely-used prototypical example of a data mining tool [190]. WEKA supports classification (e.g., decision tree learning), clustering (e.g., k-means clustering), and association rule learning (e.g., the Apriori algorithm). WEKA expects so-called “arff” files as input. Such a file stores tabular data such as shown in Tables 4.1, 4.2, and 4.3. It is impossible to directly load an event log into WEKA. However, it is possible to convert XES or MXML data into tabular data that can be analyzed by WEKA [42]. After conversion each row either corresponds to an event or a case. For example, it is possible to extract variables like flow time and the frequency of some activity for each case. Similarly, it is possible to create a table where each row lists the attributes of some event. However, either way, the original event notion is lost. This illustrates that data mining tools, like the mainstream BI products, are data-centric rather than process-centric.

Tools such as RapidMiner, KNIME, and R are extendible. For example, RapidMiner provides a marketplace where users can acquire additional building blocks (e.g., for text mining). *RapidProM* (www.rapidprom.org), available through the RapidMiner marketplace, provides a collection of process mining building blocks based on ProM plug-ins (see Sect. 11.3.3). This way users of RapidMiner are able to use process mining techniques without installing a separate process mining tool [23, 97].

In general, one cannot assume that BI and data mining tools provide any process mining capabilities. Fortunately, plenty of dedicated process mining tools are available. These are discussed in the remainder of this chapter.

11.2 Different Types of Process Mining Tools

Before describing concrete process mining tools, we first discuss different ways of characterizing process mining software.

Potentially, there may be very different groups of users interacting with process mining software. On the one hand, there may be experts that need to be able to answer “one of a kind” questions requiring ad-hoc data extractions, complex data transformations, and sophisticated analysis techniques. On the other hand, there can

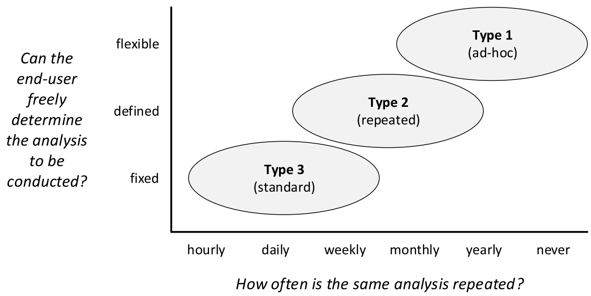

Fig. 11.1 Three types of use cases: *Type 1* (for ad-hoc questions requiring data exploration/extraction and problem-driven selection of analysis techniques), *Type 2* (for repeated questions in a known setting but possibly still requiring configuration), and *Type 3* (for standard questions in a fixed pre-configured setting)

be end-users that just want to look at standard overviews ("process-centric dashboards") generated using process mining.

The spectrum of process mining use cases can be characterized through the following two questions:

- *How often is the same analysis repeated?*
- *Can the end-user freely determine the analysis to be conducted?*

Fig. 11.1 defines three types of use cases based on answers to these two questions.

Use cases of *Type 1* (ad-hoc) require a spreadsheet-like tool: questions are ad-hoc and the user needs to have complete freedom to perform analysis. The analysis process is iterative and undefined. The results of one analysis step may lead to unanticipated additional data extractions (or transformations) to enable the next analysis step. Analysis workflows are unique and seldom repeated.

Use cases of *Type 2* (repeated) involve questions that are recurring, but at a lower frequency. Analysis workflows may be predefined but not completely fixed. Customization may be needed and the interpretation of the results requires knowledge of process mining and understanding of the data.

Use cases of *Type 3* (standard) involve routine questions that are recurring frequently. The different analysis views are fixed and no customization is possible. The user only needs to understand predefined dashboard-like views.

The three types are on the diagonal in Fig. 11.1. Use cases not on the diagonal do not make much sense. For example, we cannot provide a predefined dashboard for "one of a kind" questions (corresponding to the combination of "never" and "fixed" in Fig. 11.1). Moreover, if there is a continuous need to answer the same question based on the latest data, then there is no need to explore the data in an ad-hoc manner (i.e., also the combination of "hourly" and "flexible" in Fig. 11.1 makes no sense).

Process mining tools may be tailored to one of the three types in Fig. 11.1. For example, a tool like Disco (Fluxicon) is comparable to a spreadsheet program (but for "behavior" rather than "numbers", see Sect. 1.3). The user can load the data of interest, pick a particular view, and get immediate results without any system configuration. Such style of interaction is good for exploration and fast results (*Type 1*), but less suitable for end-users that do not understand the underlying data and analysis techniques (*Type 3*).

The initial investment for a *Type 1* analysis is low, but less suitable for situations where many users repeatedly need to do the same type of analysis. The initial investment for a *Type 3* analysis is much higher. An expert needs to configure the way data is extracted and define the views on the data provided to end users. However, after the initial investment, analysis is easier and highly repeatable. *Type 2* analysis is in-between *Type 1* and *Type 3*. Use cases of *Type 2* benefit from analysis workflows that are (partly) predefined but not completely fixed.

Another way to categorize process mining software is based on the way it is bundled:

- *Dedicated* process mining software—pure play process mining tools devoted to the analysis of event data and processes.
- *Embedded* process mining software—tools that provide process mining functionality, but that are embedded in a larger suite.

Most of the tools discussed in this chapter fall in the first category. However, process mining functionality may also be embedded in a larger BPM, ERP, BI or data mining product as an add-on. RapidProM, an extension of RapidMiner, is an example of embedded process mining software [97].

Process mining tools can also be classified based on their "openness":

- *Open-source* process mining software—the source code is publicly available. Depending on the license other parties can extend, change, or redistribute the software.
- *Closed-source* process mining software—propriety software whose source code is not published and cannot be changed or extended.

The commercial process mining tools described in Sect. 11.4 are closed-source. ProM is an example of an open-source tool.

All process mining tools are able to discover process models, but the types of models learned from event data vary. We distinguish three classes of models:

- *Informal* process models—"boxes and arrows" diagrams not having a formal interpretation that can be related to traces in the event log.
- *Formal low-level* process models—transition systems, Markov chains, episodes, sequences, etc.
- *Formal high-level* process models—end-to-end models allowing for choices, concurrency, loops, etc. This includes BPMN models, EPC models, UML activity diagrams, Petri nets, process trees, etc.

Formal models have executable semantics. Informal models are drawings composed of boxes and arrows without a clear relation to the traces in the event log. Such informal diagrams do not distinguish between choice and concurrency (there are no AND/XOR/OR-gateways/connectors/operators). A model is formal if, given a sequence of events, one can determine whether it fits or not. Process mining tools are characterized by the process models they support. Most commercial process mining tools use a mixture of informal and low-level models (see Sect. 11.4.2). The fact that a discovered model can be saved in BPMN format (or any other format with AND/XOR/OR-gateways/connectors/operators) does not imply that the model can be interpreted as such.

Process mining starts from event data. Process mining tools may have different mechanisms to get event data:

- *File*. Events are stored in a XES, MXML, Excel, or CSV file.
- *Database*. Events are loaded from a database system, for example via a JDBC connection. Several tools support incremental event loading, i.e., periodically the database is inspected for new data.
- *Adapter*. Events are loaded from a particular application (e.g., SAP, Sharepoint, or SalesForce) through a dedicated piece of software. In most cases events can be loaded incrementally.
- *Streaming*. The process mining tool works on a stream of events emitted through an event bus or web service. Events are captured as they occur and not retrieved from a file, database, or application at a later point in time.

The process mining software may run locally or remotely. The event data typically resides at the same location. We distinguish three types of deployments:

- *Stand-alone*. The software runs locally, e.g., on the laptop used for analysis.
- *On premise*. The back-end of the software does not run locally, but on a server inside the organization.
- *Cloud*. The software runs on a server outside the organization.

Some products offer multiple forms of deployment. This is not only a technological decision, but also related to privacy laws, security, and ethics. For example, the cloud provider may store event data on a server in a different country.

The refined process mining framework (Sect. 10.1) identifies the following activities:

- *Discover*—learning (process) models from event data;
- *Enhance*—repair or extend models (adding additional perspectives to a model, e.g., to show bottlenecks);
- *Diagnose*—model-based process analysis;
- *Detect*—comparing de jure models with current “pre mortem” data (events of running process instances) to detect deviations at runtime;
- *Check*—checking conformance by comparing historic “post mortem” data with de jure models (e.g., to pinpoint deviations and quantify compliance);
- *Compare*—comparing de jure models with de facto models to see whether reality deviates from what was planned or expected;
- *Promote*—transferring “best practices” (learned from event data) to the de jure model;
- *Explore*—exploring business processes at run-time using a combination of event data and models;
- *Predict*—making process-related predictions, e.g., the remaining flow time and the probability of non-compliance;

- *Recommend*—supporting operational processes by recommending suitable actions (e.g. to minimize costs or time).

Process mining software can be characterized by the activities supported. For example, all tools support activity *discover*, but only few support activities like *predict* and *recommend*.

In the remainder, we first describe ProM and then provide an overview of other process mining tools, including 11 commercial products.

11.3 ProM: An Open-Source Process Mining Platform

ProM is the leading open-source process mining tool. The lion's share of academic research is conducted by using and extending ProM. Moreover, the commercial process mining tools discussed in Sect. 11.4 are based on ideas first developed in the context of ProM. Therefore, this section first introduces ProM which is tailored towards use cases of *Type 1* (see Fig. 11.1).

11.3.1 Historical Context

In 2002, there were several, rather simple, stand-alone process mining tools available. Examples of tools developed around the turn of the century include: *MiMo* (α-miner based on ExSpect), *EMiT* (α-miner taking transactional information into account), *Little Thumb* (predecessor of the heuristic miner), *InWolvE* (miner based on stochastic activity graphs), and *Process Miner* (miner assuming structured models) [156]. At this time, several researchers were building simple prototypes to experiment with process discovery techniques. However, these tools were based on rather naïve assumptions (simple process models and small but complete data sets) and provided hardly any support for real-life process mining projects (scalability, intuitive user interface, etc.). Clearly, it did not make any sense to build a dedicated process mining tool for every newly conceived process discovery technique. This observation triggered the development of the *ProM framework*, a "plug-able" environment for process mining using MXML as input format. The goal of the first version of this framework was to provide a *common basis* for all kinds of process mining techniques, e.g., supporting the loading and filtering of event logs and the visualization of results. This way people developing new process discovery algorithms did not have to worry about extracting, converting, and loading event data. Moreover, for standard model types such as Petri nets, EPCs, and social networks, default visualizations were provided by the framework.

In 2004, the first fully functional version of the ProM framework (*ProM 1.1*) was released. This version contained 29 plug-ins: 6 mining plug-ins (the classic α-miner, the Tshinghua α miner, the genetic miner, the multi-phase miner, the social network miner, and the case data extraction miner), 7 analysis plug-ins (e.g., the

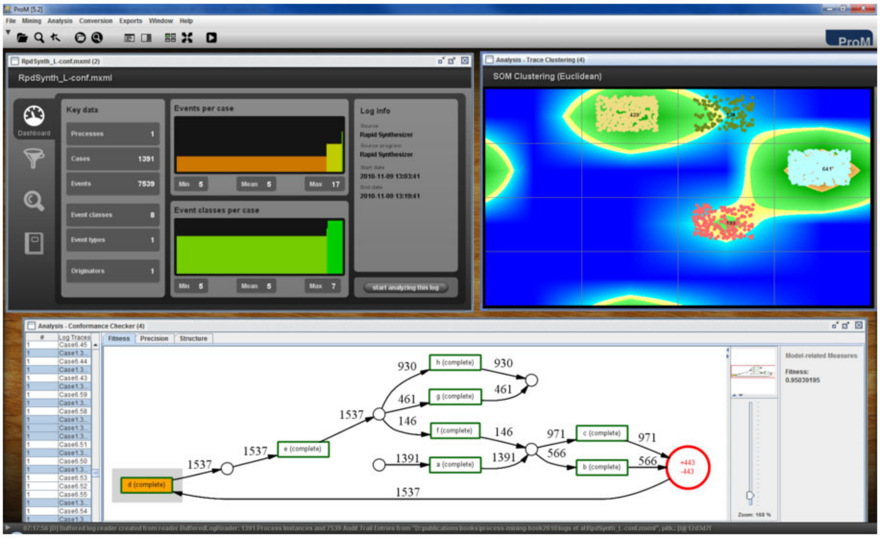

Fig. 11.2 Screenshot of ProM 5.2 showing two of the 286 plug-ins. The *bottom window* shows the *conformance checker* plug-in while checking the fitness of event log L_{full} described in Table 8.1 and WF-net N_2 depicted in Fig. 8.2. The plug-in identifies the conformance problem (the log and model disagree on the position of d) and returns a fitness value computed using the approach presented in Sect. 8.2, $\mathit{fitness}(L_{full}, N_2) = 0.95039195$. The *right window* shows the trace clustering plug-in using Self Organizing Maps (SOM) to find homogeneous groups of cases. The largest cluster contains 641 cases. These are the cases that were rejected without a thorough examination (i.e., traces σ_1, σ_3, σ_{13} in Table 8.1)

LTL checker), 4 import plug-ins (e.g., plug-ins to load Petri nets and EPCs), 9 export plug-ins, and 3 conversion plug-ins (e.g., a plug-in to convert EPCs into Petri nets). Over time more plug-ins were added. For instance, *ProM 4.0* (released in 2006) contained already 142 plug-ins. The 27 mining plug-ins of ProM 4.0 included also the heuristic miner and a region-based miner using Petrify. Moreover, ProM 4.0 contained a first version of the conformance checker described in [121]. *ProM 5.2* was released in 2009. This version contained 286 plug-ins: 47 mining plug-ins, 96 analysis plug-ins, 22 import plug-ins, 45 export plug-ins, 44 conversion plug-ins, and 32 filter plug-ins. Figure 11.2 shows two plug-ins of ProM 5.2. This version already supported most of the process mining techniques presented in this book. For example, the 47 mining plug-ins of ProM 5.2 include most of the discovery algorithms presented in Chap. 7 (genetic mining, heuristic mining, fuzzy mining, etc.). The replay approach presented in Sect. 8.2 was supported by the conformance checker plug-in of ProM 5.2 [121].

The spectacular growth of the number of plug-ins in the period from 2004 to 2009 illustrates that ProM realized its initial goal to provide a platform for the development of new process mining techniques. ProM had become the de facto standard for process mining. Research groups from all over the globe contributed to the development of ProM and people from tens of thousands of organizations downloaded ProM (the ProM framework has been downloaded over 130.000 times). In the same period, we applied ProM at numerous organizations, e.g., in the context of joint research projects, Master projects, and consultancy projects. The large number of plug-ins and the many practical applications also revealed some problems. For example, ProM 5.2 can be quite confusing for the inexperienced user who is confronted with almost 300 plug-ins. Moreover, in ProM 5.2 (and earlier versions) the user interface and the underlying analysis techniques are tightly coupled, i.e., most plug-ins require user interaction. It was impossible to embed ProM functionality in data mining tools such as RapidMiner, KNIME, etc. due to this tight coupling.

To be able to run ProM remotely and to embed process mining functionality in other systems, we decided to completely re-implement ProM from scratch. This allowed us to learn from earlier experiences and to develop a completely new architecture based on an improved plug-in infrastructure.

ProM 6 was released in November 2010. This was the first version based on the new architecture and XES rather than MXML. XES, described in Sect. 5.3, is the process mining standard adopted by the IEEE Task Force on Process Mining. Although ProM 5.2 was already able to load enormous event logs, scalability and efficiency were further improved by using OpenXES [64, 65]. Not all plug-ins of ProM 5.2 have been re-implemented in ProM 6. Nevertheless, most of the process mining techniques described in this book are supported by plug-ins developed for ProM 6.

ProM 6 can distribute the execution of plug-ins over multiple computers. This can be used to improve performance (e.g., using grid computing) and to offer ProM as a service. For instance, at TU/e (Eindhoven University of Technology) we use a dedicated process mining grid to handle huge data sets and to conduct large-scale experiments. The user interface has been re-implemented to be able to deal with many plug-ins, logs, and models at the same time. Plug-ins are now distributed over so-called *packages* and can be chained into composite plug-ins. Packages contain related sets of plug-ins. ProM 6 provides a so-called package manager to add, remove, and update packages. Users should only load packages that are relevant for the tasks they want to perform. This way it is possible to avoid overloading the user with irrelevant functionality. Moreover, ProM 6 can be customized for domain-specific or even organization-specific applications.

Figures 11.3 and 11.4 show the selection of the ILP miner plug-in (based on language-based regions, see Sect. 7.4.3) and the resulting process model discovered by ProM 6. ProM 6.5.1a (SilvR+) was released in October 2015. There is also a "ProM Lite" version providing only the most used functionality.

11.3.2 Example ProM Plug-Ins

ProM is open-source software[1] and can be freely downloaded from www.promtools.org or www.processmining.org. Plug-ins can be installed via ProM's package man-

[1] ProM framework is released under the GNU Lesser General Public License (L-GPL).

Fig. 11.3 Screenshot of ProM 6.5. After loading an event log, a list of applicable plug-ins is shown and the plug-in implementing discovery using language-based regions (ILP miner) is selected

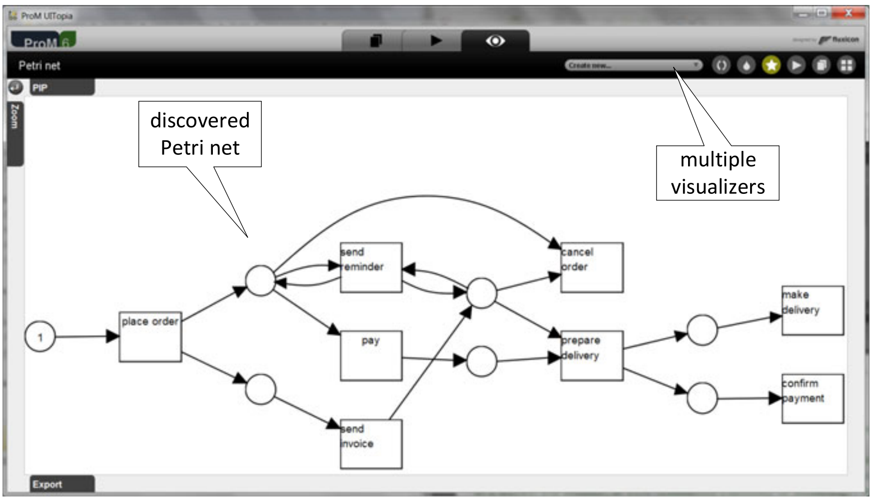

Fig. 11.4 Screenshot of ProM 6.5 showing the Petri net discovered using language-based regions after starting the plug-in selected in Fig. 11.3

ager. Currently, there are over 1500 plug-ins available (including deprecated plug-ins that are no longer supported). The ILP miner plug-in depicted in Fig. 11.4 is just one of these 1500 plug-ins. Hence, it is impossible to provide a complete overview of the functionality of ProM. The reader is encouraged to visit www.processmining.org to learn more about ProM's functionality and available plug-ins. Here, we only show a few examples.

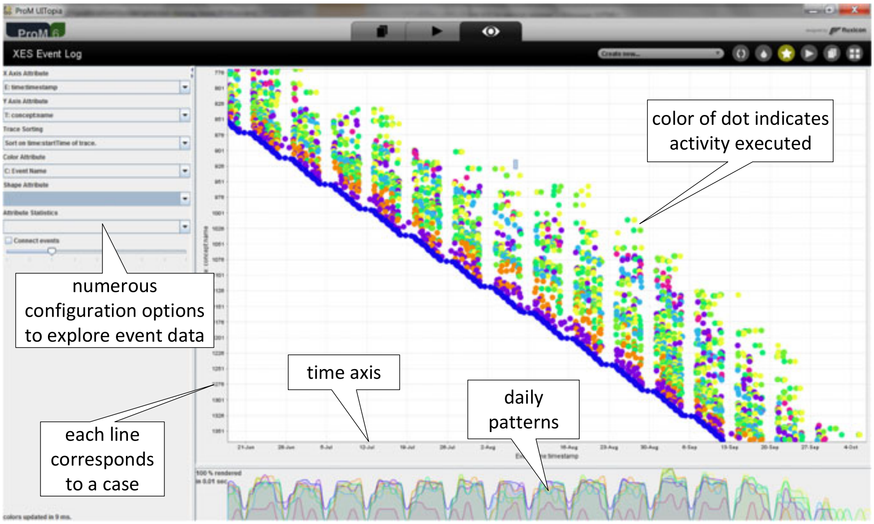

Fig. 11.5 ProM's dotted chart can be used to explore the event data from different angles

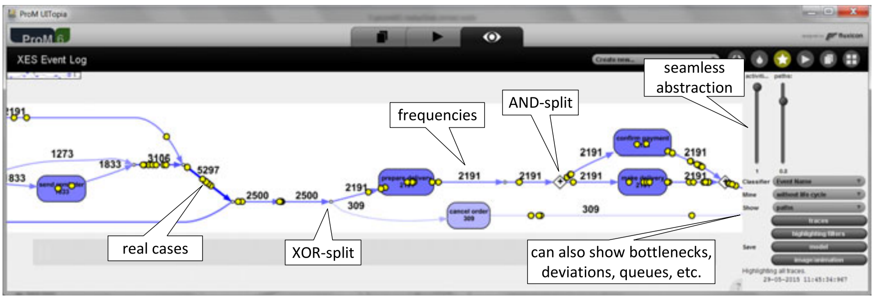

Fig. 11.6 Visual inductive miner replaying the event log on the discovered process model

ProM can load XES, MXML, and CSV files. To extract files from other data sources, tools such as XESame and ProMimport can be used (cf. Sect. 5.3). Figure 11.5 shows a *dotted chart* (see Sect. 9.2). The user can control both axes completely and influence the coloring and shape of the dots.

ProM supports dozens of process discovery algorithms. Next to the ILP miner shown in Fig. 11.4 and the α-algorithm [157], also heuristic mining [183, 184], fuzzy mining [66], genetic process mining [12, 26], and various forms of inductive mining [89–91] are supported. Figure 11.6 shows the *visual inductive miner*. This miner always returns a sound process model and is able to handle large and noisy event logs. Nevertheless, the miner can ensure (if desired) perfects fitness. Results can be converted to Petri nets, EPCs, statecharts and BPMN models. Moreover, the visual inductive miner supports bottleneck analysis and outlier detection.

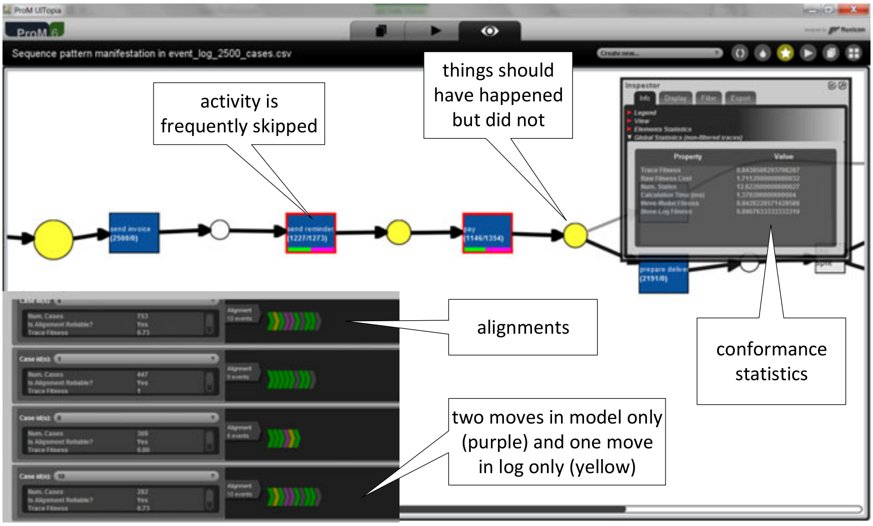

Fig. 11.7 Conformance checking based on alignments (cf. Sect. 8.3)

Conformance checking and performance analysis heavily depend on reliable replay algorithms [169]. Newer plug-ins in ProM rely on alignments as described in Sect. 8.3. Figure 11.7 shows deviations from both the log and model perspective. Next to fitness also notions such as precision are computed [5]. Similar views are provided for performance diagnostics based on alignments.

ProM also supports trace clustering [78], trace alignment [79], and model repair [52]. Next to a variety of procedural models (Petri nets, BPMN, YAWL, EPCs, etc.), ProM also support declarative models. Declare models can be discovered and the conformance of declarative models can be checked.

ProM is not limited to the control-flow and time perspectives. There are plug-ins to create social networks and to discover roles (organizational perspective). There are also plug-ins for decision mining [40, 120]. These plug-ins enhance control-flow models with guards based on the data perspective. Plug-ins can discover so-called *data-aware Petri nets* and check the conformance of such models [40]. Many of these plug-ins create classification problems. See [42] for a ProM plug-in that supports the interaction between process mining and data mining in a generic manner.

The plug-ins mentioned thus far are all related to process mining. However, it should be noted that ProM (both version 5.2 and 6.X) supports process analysis in the broadest sense, e.g., also the analysis techniques mentioned in Sect. 3.3 are supported by ProM or the tools that ProM interfaces with (e.g., CPN Tools). For example, the plug-in "Analyze structural properties of a Petri net" computes transition invariants, place invariants, S-components, T-components, traps, siphons, TP- and PT-handles, etc. The plug-in "Analyze behavioral properties of a Petri net" computes unbounded places, dead transitions, dead markings, home markings, coverability graphs, etc. The "Woflan" plug-in checks the soundness of WF-nets (cf. Sect. 3.2.3) [179]. Moreover, powerful Petri-net-based analysis tools such as *LoLa*, *Wendy*, *Uma*, and *Petrify* are embedded in ProM as plug-ins.

The hundreds of ProM plug-ins implementing all of the techniques described in this book (and many more) illustrate the applicability and broadness of process mining.

11.3.3 Other Non-commercial Tools

Due to the success of ProM in the academic community, there are only a few other non-commercial process mining tools. Research groups all over the world have contributed to the 1500 plug-ins in ProM (also see the list of organizations mentioned in the Acknowledgements). Next to ProM, most other tools are commercial (cf. Sect. 11.4). A few notable exceptions are described next.

11.3.3.1 PMLAB

PMLAB is a scripting environment for process mining developed by the group of Josep Carmona at Universitat Politècnica de Catalunya in Barcelona. PMLAB can load XES, MXML, and CSV files and different analysis steps can be chained together in scripts. A variety of process discovery techniques based on the theory of regions and satisfiability modulo theories are supported. Tools such as *Genet*, *Petrify*, *Rbminer*, and *Dbminer* can be invoked from PMLAB. These tools are mostly based on state-based regions [34]. As shown in Sect. 7.4, an event log can be converted into a transition system and subsequently synthesized into a Petri net. Classical region theory needs to be extended/relaxed to make it more applicable for process discovery, e.g., Rbminer adapts the classical theory to provide more compact and readable process models [128]. PMLAB uses iPython, a framework for scripting in Python. PMLAB can also invoke ProM plugins through PMLAB scripts. The scripting tool was inspired by MATLAB and Mathematica. However, there are also similarities with RapidMiner, KNIME, and R. PMLAB can be downloaded from https://www.cs.upc.edu/~jcarmona/PMLAB/.

11.3.3.2 CoBeFra

CoBeFra is a benchmarking framework for conformance checking developed at the department of Management Informatics at KU Leuven in Belgium. It is mostly used for the systematic evaluation of process discovery techniques. CoBeFra reads event logs in XES or MXML file format and process models in PNML format. Given an event log and a process model, the tool evaluates the model with respect to dozens of metrics (e.g., various notions of fitness, precision, and simplicity). For large scale experiments (e.g. varying parameters of discovery algorithms to analyze the effects on fitness and precision), the metrics are automatically collected for sets of models and logs. CoBeFra can be downloaded from http://www.processmining.be/cobefra.

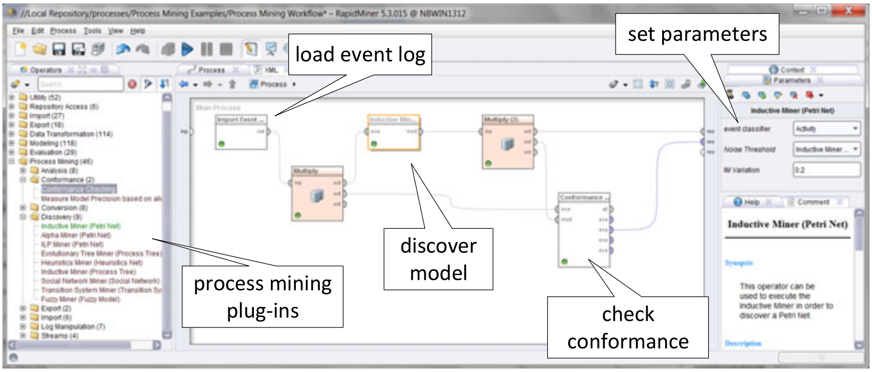

Fig. 11.8 A process mining workflow where an event log is loaded, a model is discovered using the inductive miner, and the result is checked using the conformance checker based on alignments

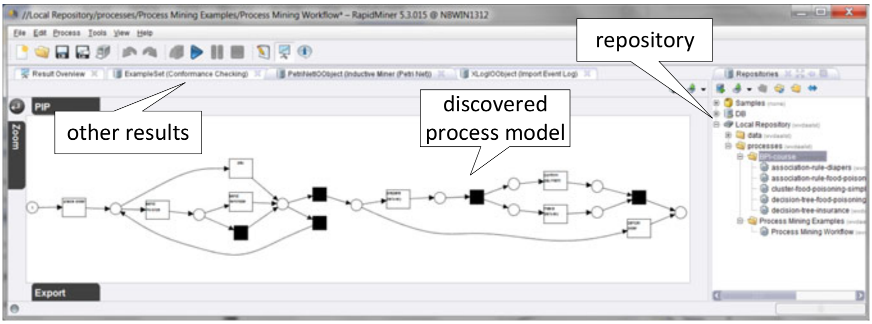

Fig. 11.9 One of the outcomes of the analysis workflow in Fig. 11.8

11.3.3.3 RapidProM

Many tools for data analysis support the definition and execution of *analysis workflows*, sometimes also called *scientific workflows*. For example, widely used tools like RapidMiner, KNIME, and R can chain together building blocks to form such workflows. However, these tools do not support process mining natively. Conversely, ProM does not provide such workflow support. Therefore, *RapidMiner* was extended with process mining plug-ins from ProM. The resulting tool is called *RapidProM* (www.rapidprom.org).

Figure 11.8 shows a process mining workflow created using RapidProM. First, a XES log is loaded. Second, a process model is discovered using the *Inductive Miner—infrequent* (IMF, [89]). The quality of this model is checked using alignments using another building block. The workflow in Fig. 11.8 can be stored and applied to any event log. Figure 11.9 shows one of the output objects produced by the workflow (the discovered process tree was automatically converted to a Petri net).

RapidProM can be used to do large scale experiments [23, 97]. For example, the workflow in Fig. 11.8 can be applied to thousands of event logs without any manual intervention. RapidProM can also be applied to answer recurring questions in a business setting, e.g., to create a report at the end of every week.

RapidProM is available via the *RapidMiner Marketplace*. Many other types of analysis are available in the RapidMiner ecosystem. This facilitates the combination of process mining, text mining, machine learning, data mining, and statistics. For example, cases can be grouped into clusters using standard data mining techniques followed by the application of process mining techniques on each of these clusters.

Whereas ProM is most suitable for use cases of *Type 1*, RapidProM, CoBeFra, and PMLAB are tailored towards use cases of *Type 2* (see Fig. 11.1).

11.4 Commercial Software

Several commercial process mining tools emerged on the market in recent years. Compared to ProM these tools are easier to use, but provide less functionality than the 1500 plug-ins in ProM. This lowers the threshold for using process mining significantly.

This section provides an overview of the commercial process mining tools currently on the market. *The goal is not to give detailed information on specific tools or to provide checklists.* The market and tools change rapidly. Most of the tools described did not exist when the first version of this book was published in 2011 [140]. Moreover, the capabilities of tools change with every release and usability and scalability cannot be expressed in simple checklists. *Hence, organizations that are selecting a commercial process mining tool are urged to evaluate the tools based on concrete questions and datasets.*

After illustrating some of the commercial tools in Sect. 11.4.1, we share a few general insights based on experiences with currently available process mining tools in Sect. 11.4.2.

11.4.1 Available Products

Table 11.1 lists 11 process mining tools in alphabetical order: *Celonis Process Mining* (Celonis), *Disco* (Disco), *Enterprise Discovery Suite* (EDS), *Interstage Business Process Manager Analytics* (Fujitsu), *Minit* (Minit), *myInvenio* (myInvenio), *Perceptive Process Mining* (Perceptive), *QPR ProcessAnalyzer* (QPR), *Rialto Process* (Rialto), *SNP Business Process Analysis* (SNP), and *webMethods Process Performance Manager* (PPM). For tools with a longer name, the shorter name between brackets is used. For example, "webMethods Process Performance Manager" is abbreviated to PPM.

Tools like Disco, Fujitsu, QPR, and PPM have been around for a few years. Minit, myInvenio, and Rialto emerged very recently (in 2015). Tools like *Process*

Table 11.1 Overview of commercial process mining tools

Short name	Full name of tool	Version	Vendor	Webpage
Celonis	Celonis Process Mining	4.0	Celonis GmbH	www.celonis.de
Disco	Disco	1.9	Fluxicon	www.fluxicon.com
EDS	Enterprise Discovery Suite	4	StereoLOGIC Ltd	www.stereologic.com
Fujitsu	Interstage Business Process Manager Analytics	12.2	Fujitsu Ltd	www.fujitsu.com
Minit	Minit	1.0	Gradient ECM	www.minitlabs.com
myInvenio	myInvenio	1.0	Cognitive Technology	www.my-invenio.com
Perceptive	Perceptive Process Mining	2.7	Lexmark	www.lexmark.com
QPR	QPR ProcessAnalyzer	2015.5	QPR	www.qpr.com
Rialto	Rialto Process	1.5	Exeura	www.exeura.eu
SNP	SNP Business Process Analysis	15.27	SNP Schneider-Neureither & Partner AG	www.snp-bpa.com
PPM	webMethods Process Performance Manager	9.9	Software AG	www.softwareag.com

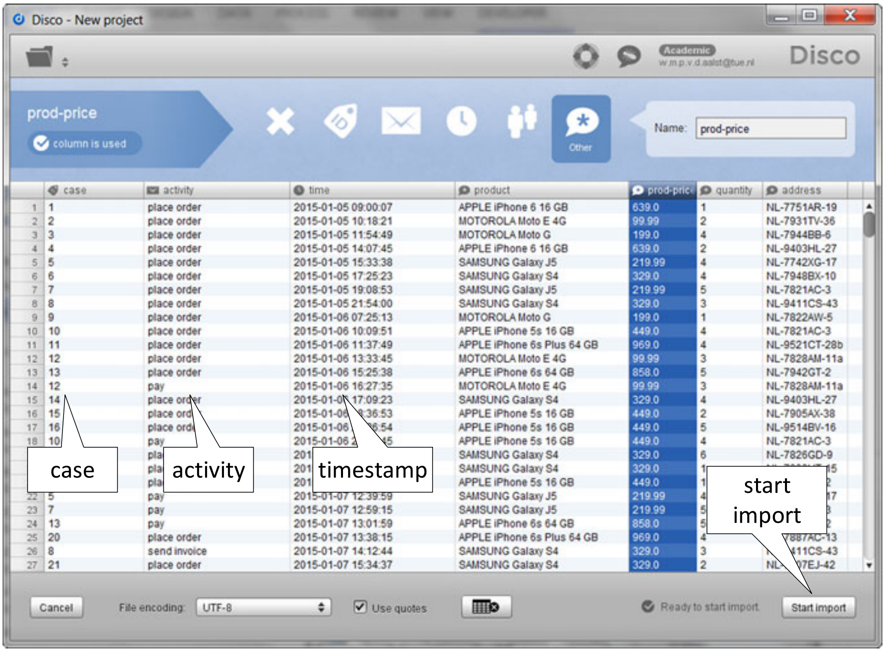

Fig. 11.10 Disco allows for the easy import of CSV files and supports process mining formats such as XES and MXML

Discovery Focus (Iontas/Verint Systems) and *Enterprise Visualization Suite* (Businesscape) are no longer available. Earlier products such as *Reflect|one* by Pallas Athena and *Reflect* by Futura Process Intelligence where further developed as part of the Perceptive suite of BPM tools. It is interesting to note that both Pallas Athena and Futura Process Intelligence were selected as "Cool Vendor" by Gartner in 2009 because of their process mining capabilities. Reflect was the first dedicated commercial process mining tool. The *ARIS Process Performance Manager* (PPM) was initially developed by IDS Scheer. Process mining capabilities were added later and PPM is now part of Software AG's webMethods Operational Intelligence Platform.

As mentioned, it is not our goal to discuss particular tools in detail. However, we show a few screenshots to provide an impression of typical capabilities of available tools. Figure 11.10 shows a screenshot of Disco while loading a CSV file. The columns can be mapped onto process mining concepts such as case, activity, timestamp, and resource. Disco automatically suggests an initial mapping (including the format to be used for timestamps) that can be adapted. Disco can also load and save event logs in XES and MXML format. Researchers often use Disco for an initial analysis of the data (involving filtering, exploration, and bottlenecks analysis) after which XES files are saved for further analysis using ProM. The discovery algorithm used by Disco can be viewed as an improved and further developed version of ProM's Fuzzy Miner [66]. The scalability and robustness of Disco are much better than the original Fuzzy Miner. Disco is easy to use and learn, and lowers the barrier

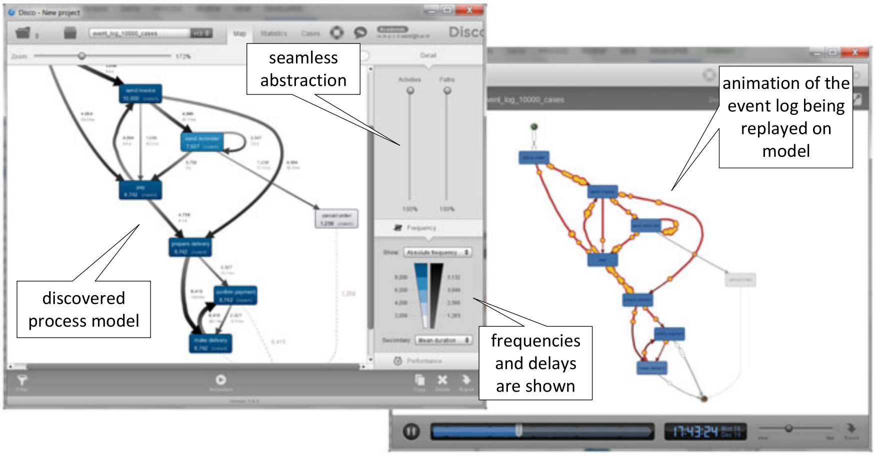

Fig. 11.11 The discovery algorithm of Disco is a further development of the Fuzzy miner and event data can be replayed at the selected abstraction level

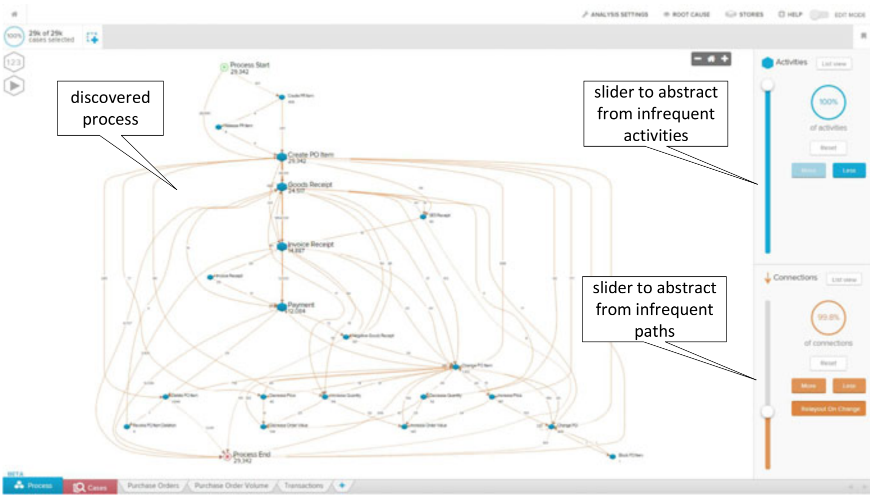

Fig. 11.12 A process model discovered using Celonis showing all activities in the event log

to get started with process mining significantly. Figure 11.11 shows a discovered process model and an animation based on the underlying event data. Animations can be saved as movies and show behavior that changes over time.

Figure 11.12 shows a process model discovered using Celonis. Celonis can load event data from CSV and XES files or database management systems such as SAP HANA, Oracle DB, MSSQL, MYSQL, PostgreSQL and IBM DB2. It is often used in conjunction with SAP. Events are stored in an OLAP-like data structure. Like Disco, Celonis provides sliders to seamlessly simplify models (if desired). In

Fig. 11.13 Animation of the process obtained by replaying the event data on a simplified process model and three charts showing trends in the event data

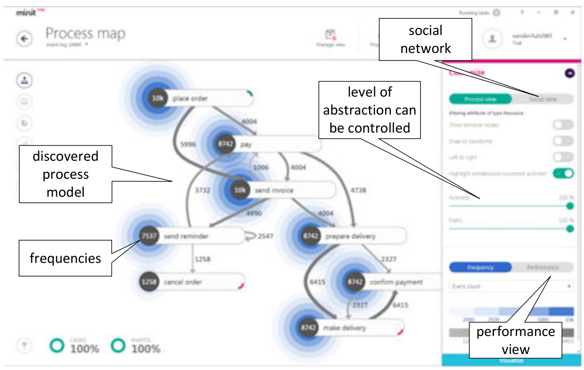

Fig. 11.14 A process model discovered using Minit

Fig. 11.13, a simplified model is used to show an animation of the process. Process related information can also be summarized in column-, line-, area-, pie-charts or tables. This is illustrated by the three charts in Fig. 11.13.

Figure 11.14 shows a screenshot of a model discovered using Minit. Minit also supports XES and uses a discovery algorithm similar to ProM's Fuzzy Miner. Like Disco and Celonis, Minit is able to handle large event logs efficiently.

Like Minit, myInvenio became available in 2015. These tools illustrate the growing interest in process mining. Figure 11.15 shows a process model discovered using myInvenio. Conformance checking is supported by comparing a reference model (e.g., specified in BPMN or XPDL) with the discovered process model. The differences can be highlighted as shown Fig. 11.15.

Figure 11.16 shows a process model discovered using Perceptive Process Mining. Performance-related information is mapped onto the process model (durations and

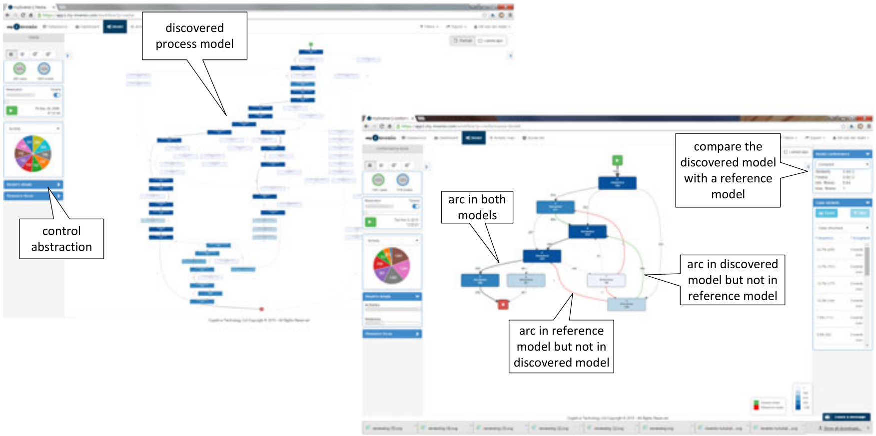

Fig. 11.15 A process model discovered using myInvenio (*left*) and the comparison of a reference model and a discovered model (*right*)

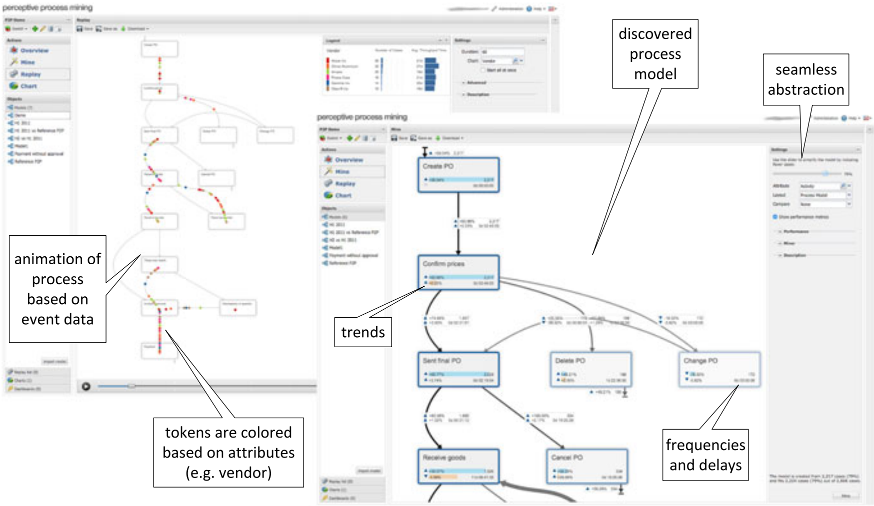

Fig. 11.16 The process model discovered using Perceptive is used to signal trends in performance (*right*) and to animate the process using "colored tokens"(*left*)

frequencies). Perceptive also shows trends, e.g., bottlenecks that are growing over time. The animation in Fig. 11.16 uses colored tokens. The coloring can be based on any case attribute (e.g., the vendor). This helps to spot differences between distinct groups of cases.

Figure 11.17 shows screenshots of four other process mining tools. Each process shown was discovered using event data.

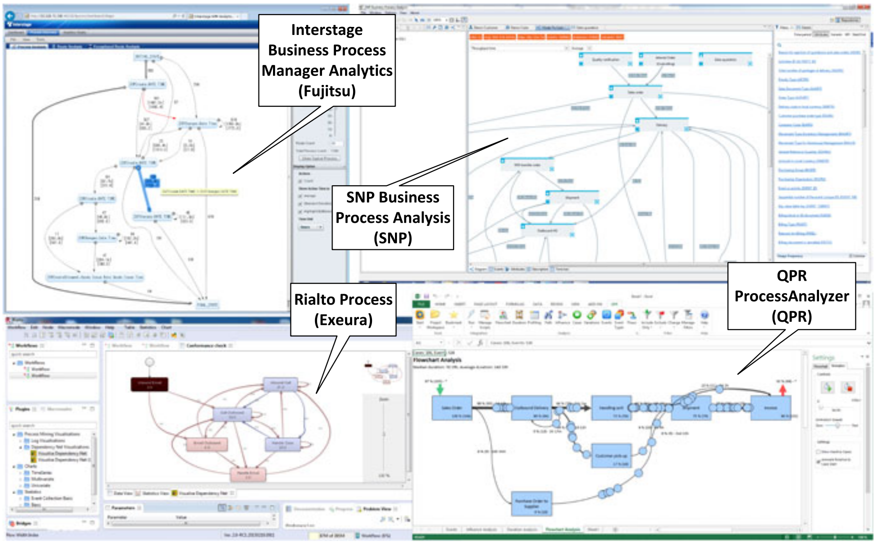

Fig. 11.17 Screenshots of four additional process mining tools: Fujitsu Interstage Business Process Manager Analytics (Fujitsu), SNP Business Process Analysis (SNP), QPR ProcessAnalyzer (QPR), and Exeura Rialto Process (Rialto)

11.4.2 Strengths and Weaknesses

The screenshots in Figs. 11.10–11.17 show that process mining capabilities are readily available in commercial tools. None of the products covers the range of process mining capabilities supported by the hundreds of available ProM plug-ins. However, ProM requires process mining expertise and is not supported by a commercial organization. Hence, it has the advantages and disadvantages common for open-source software. Fortunately, the 11 process mining tools listed in Table 11.1 nicely complement ProM.

On the one hand, there are quite some commonalities among the commercial tools (as illustrated by the screenshots in Figs. 11.10–11.17). On the other hand, there are major differences in usability and scalability. Some focus more on use cases of *Type 1* (e.g., Disco, Minit, and myInvenio) whereas other tools focus more on use cases of *Type 3* (e.g., Celonis and PPM). Organizations selecting a commercial process mining product are urged to do a pilot project where a few products are applied to organization-specific data and questions.

Despite the differences between the tools, we can make some general observations.

11.4.2.1 Limited Support for Concurrency

If a group of activities is not always executed in the same order, we would like to avoid the situation where all activities are connected to one another. Yet, process

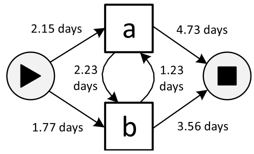

Fig. 11.18 Most tools that do not allow for concurrency have difficulties handling event logs like $L_{par} = [\langle a, b\rangle^{100}, \langle b, a\rangle^{100}]$: Due to the introduction of loops, the model allows for non-observed traces like $\langle a, b, a\rangle$, $\langle a\rangle$, $\langle b, a, b, a, b, a, b\rangle$, $\langle b\rangle$, etc.

discovery techniques that do not support concurrency do exactly that. In Sect. 11.2, these purely sequential models were called low-level models. The models discovered by such techniques tend to be very Spaghetti-like. Moreover, sequential models where every activity appears only once tend to be severely underfitting (e.g., parallel activities are turned into loops).

To illustrate the problem consider the artificial event log $L_{par} = [\langle a, b\rangle^{100}, \langle b, a\rangle^{100}]$. Clearly, there is no loop and one would expect that the discovered model shows that a and b both happen once per case. However, if a and b cannot be concurrent and the tool has one node per activity, then the tool is forced to introduce loops allowing for traces like $\langle a, b, a, b, a\rangle$ (see Fig. 11.18). Clearly, this model is underfitting and not adequately reflecting the observed behavior.

Some of the commercial tools do not support concurrency at all (e.g., SNP). Perceptive Process Mining offers two mining algorithms: a genetic algorithm able to discover concurrency (but time-consuming and not scalable) and a simpler, better performing, algorithm based on the directly follows relation having the problem mentioned above.

Also Disco deals with concurrency different from the algorithms described in Chaps. 6 and 7. Parallelism in Disco is discovered only if two activity instances for the same case overlap. This implies that concurrency cannot be discovered in event logs without explicit transactional information (e.g., when there are just complete events). If activities are interleaved (i.e., not overlapping), then the arrows are suppressed, unless the slider is moved up to ensure perfect fitness. Using the terminology introduced in Sect. 11.2, Disco shifts from an informal model with concurrency to a formal low-level model without concurrency to ensure correctness.

Other tools have similar issues and are often less clear about this. They operate in the space between informal models and low-level models, thus making interpretation tricky. Consider the delays in Fig. 11.18. When does the process end—4.73 days after the (last) completion of activity a or $2.23 + 3.56 = 5.79$ days after the (last) completion of activity a? Probably none of the two answers is right, thus illustrating the confusion.

To summarize—*None of the commercial tools handles concurrency adequately.* There are at least two reasons for this. First of all, simple algorithms are used to ensure scalability and transparency. Second, the models learned have informal semantics. The latter is interesting because several tools claim to support BPMN and can export models to BPM systems. This may be lead to misleading results. Most tools do not show explicit AND/XOR-splits/joins. Adding logic when saving models will lead to confusion and may result in models that are not sound (e.g., having deadlocks).

As long as process models are interpreted as "pictures" this is not a problem. However, the way that models need to be interpreted *also influences frequencies and performance results*. For example, if it is unclear whether things need to be synchronized or not, computed waiting times cannot be trusted. The inductive mining techniques presented in Sect. 7.5 show that it is possible to discover concurrency without creating unsound or imprecise models. The different inductive mining algorithms (IM, IMF, IMC, IMD, IMFD, IMCD, etc.) always produce sound models and are highly scalable. Some of these algorithms even provide formal guarantees (e.g., perfect fitness).

11.4.2.2 Limited Support for Conformance Checking

Informal models that can only be interpreted as "pictures" cannot be used for conformance checking. Currently, there is no commercial tool that computes alignments or that is able to apply some other replay algorithm to precisely diagnose deviations in the presence of concurrency. The reasons are the same as before: scalability (computing alignments may be too time consuming) and informal semantics (e.g., not being able to distinguish between AND-joins and XOR-joins).

Conformance checking is not handled by replaying the event log on a precise end-to-end process model. Instead one or more of the following approaches are used:

- *Rule based*. The user can specify rules for filtering. For example, Disco and Celonis can be used to define a wide variety of rules (e.g., "a is followed by b and not c and should be executed by a resource not involved in d"). By applying such rules, the event log can be split into conforming and non-conforming cases.
- *Outlier based*. Infrequent paths that deviate from mainstream behavior are manually inspected. By classifying certain paths as deviating, the corresponding cases are tagged as non-conforming.
- *Side-by-side*. The discovered process model and the normative reference model are depicted next to each other. Users need to visually compare models to see deviations.
- *Overlay*. The discovered model and the reference model are stacked on top of each other and differences are highlighted. Figure 11.15 illustrates that myInvenio supports this type of comparison.

Comparing a discovered model and a reference model may lead to incorrect conclusions. Note that the discovered model generalizes over the data, i.e., paths possible in the model may never have happened. This may trigger the detection of deviations that never occurred in reality. The discovered model may also abstract from infrequent behavior. Therefore, rare (but possibly harmful) deviations may remain undetected. However, such peculiar deviations tend to be highly relevant for conformance checking.

Assume that the informal model in Fig. 11.18 was discovered for L_{par} and subsequently compared with a normative model putting a and b in parallel. A visual comparison of the two models would suggest non-existing deviations.

The techniques in Chap. 8 and the plug-ins of ProM show that conformance checking is possible. However, conformance checking can only be supported if the informal models are replaced by formal models (e.g., process trees or Petri nets with a defined initial and final marking). *As long as this functionality is not present, users are forced to capture the real semantics of the normative model in terms of a collection of rules used for filtering.*

11.4.2.3 Performance Perspective is Well Supported

The primary focus of commercial process mining tools is on performance. Each of the tools can visualize bottlenecks in the process. Tools such as Celonis, Perceptive, QPR, Minit, PPM, etc. provide a range of charts. Most of the commercial tools make it possible to quickly find bottlenecks, unnecessary rework and delays.

Note that the problems mentioned earlier may endanger the correctness of performance results. If misalignments and concurrency are not handled well, then the reported results may be incorrect. For example, tools may report negative waiting times if events are reversed or excessive times if events are missing.

11.4.2.4 Data Perspective Not in Models

None of the commercial process mining tools is able to discover data-aware process models. For example, it is impossible to learn guards or perform any other form of decision mining as described in Chap. 9. Conformance checking of models with data is also not supported.

Additional data in the event log can be used in rules for filtering. Moreover, some tools can show the distribution of values for particular groups of cases. However, data are not explicitly related to the process model.

11.4.2.5 Organizational Perspective

Most tools are able to construct a social network (see Chap. 9). It is typically also possible to see the utilization of resources. Nearly all tools consider resource information, roles, and other organizational entities as plain data elements. Hence, the organizational perspective can be handled in the same way as data (e.g., using filtering). Separation of duties (4-eyes principle) can be checked in this way. myInvenio also creates an activity map (a simplified RACI matrix). Normally, a RACI matrix shows the people Responsible, Accountable, Consulted, and Informed for each activity. Event logs need to be enriched to provide this type of information. Sophisticated analysis techniques to optimize work distribution and social network analysis are still missing.

11.4.2.6 Growing Support for XES

Next to tools like ProM, RapidProM, PMLAB, and CoBeFra, the XES standard is supported by a growing number of commercial tools. Currently, Disco, Celonis, Minit, Rialto, and SNP support XES. QPR and myInvenio have announced XES support for the next release. Perceptive, PPM, EDS, and Fujitsu do not (yet) support XES.

XES makes it easier to combine different tools, e.g., using a commercial tool in conjunction with ProM, RapidProM, PMLAB, or CoBeFra.

11.4.2.7 Getting Event Data from Other Sources

Vendors of commercial tools realize that substantial time is spent on extracting data from information systems. In Sect. 11.2, we listed four mechanisms to get event data: file, database, adapter, and streaming. Next to file-based imports of XES, MXML, and CSV, most tools support the extraction of data from JDBC databases. Events can often be loaded from systems such as MySQL, IBM DB2, Oracle DB, SQL Server, PostgreSQL, and SAP HANA.

Often datasets can also be *incrementally* updated (importing only changes since the last import). For example, Disco can retrieve data from a server with the so-called Airlift interface. On the server side of the Airlift connection, arbitrary databases and production systems can be connected.

Systems like Celonis provide additional support to obtain data from SAP systems. Due to the partnership between SAP and Celonis, integration with SAP products like SAP HANA is safeguarded. In fact, most process mining tools support application specific adapters, but the range of systems covered and the quality of these adapters varies per tool.

11.4.2.8 Filtering

Filtering plays a crucial role in most commercial systems. Figure 11.19 shows six types of filtering supported by Disco. Filters can be used to remove individual events or complete cases. For example, one can remove all slow cases, all exceptional cases, etc. One can also specify LTL or Declare-like rules, e.g., activity a should be eventually followed by b (the response constraint in Declare and "$\Box(a \Rightarrow (\Diamond b))$" in LTL). Filtering can be used for ad-hoc conformance checking and plays an important role in root-cause analysis.

Filtering is related to OLAP (see Sect. 12.4). The dimensions in an OLAP cube also split the data based on different criteria. Process mining tools like Celonis store events in multidimensional cubes to facilitate the selection and comparison of particular groups of cases.

11.4.2.9 No Automatic Clustering

Filtering and the selection of dimensions in an OLAP cube are based on user-defined criteria. However, one may also use clustering techniques that automatically group

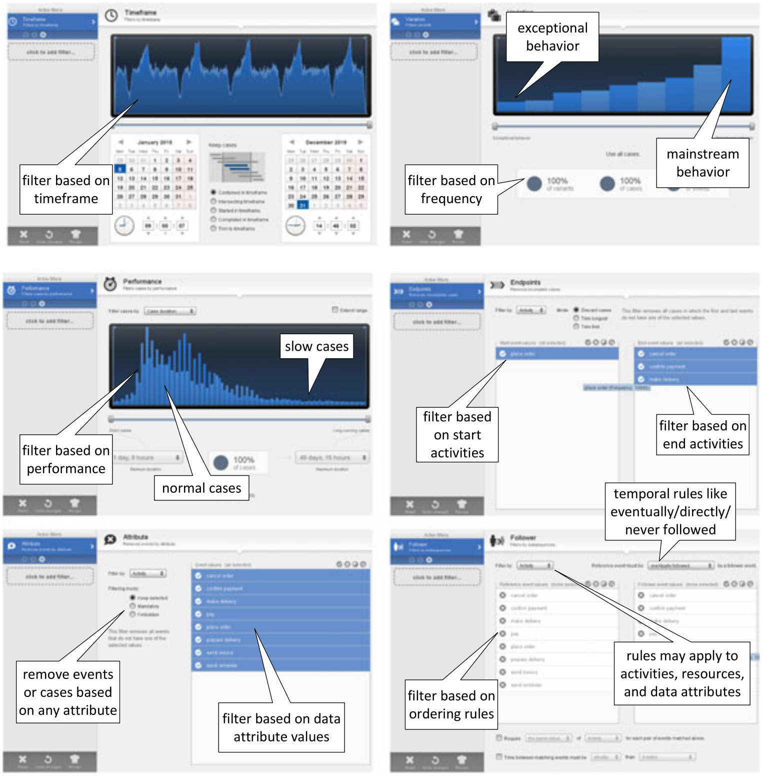

Fig. 11.19 Illustration of the extensive filtering capabilities in commercial systems like Disco

cases that are similar. ProM provides several ways of clustering similar cases based on selected features.

Standard techniques like k-means clustering (see Sect. 4.3 and Chap. 9) can be used as a preprocessing step for process mining [13, 62, 78]. The clusters themselves may already provide novel insights. Moreover, the clusters can often be used to discover multiple simple process models instead of one complex process model.

Surprisingly, clustering is not supported by the current generation of commercial process mining tools.

11.4.2.10 Reporting and Animation

Process mining results need to be communicated. Most tools provide means to create reports, for example, by storing artifacts such as charts, tables, and models. Compared to BI tools, the reporting facilities of process mining tools are often limited.

Disco, Celonis, Perceptive, Minit, QPR, and myInvenio support data-driven process animations. Figures 11.6, 11.11, 11.13, and 11.16 show screenshots where event logs are animated by replaying them on discovered models. Such animations are instrumental when convincing management. Animation is also a means to support change management: It can be used to create a sense of urgency and to build consensus on root causes.

11.4.2.11 Links to Other Tools

Some of the process mining tools are part of a bigger suite. For example, PPM is part of the webMethods suite and the ARIS family of tools. Perceptive Process Mining was developed as part of Perceptive's BPM suite. It is expected that in time most BPM systems will provide a process mining component (similar to the simulation components in today's BPM systems).

Most process mining tools are able to export process models in a format that can be read by other tools (e.g., BPMN or XPDL). This way the results from process mining can be used as a starting point for modeling, simulation, and documentation.

As mentioned in the context of RapidProM, the interplay between process mining and data mining is extremely valuable. Hence, some process mining tools can export data in a form that can be analyzed by standard data mining tools. Other crossovers of tools are possible. For example, loading process mining results into Excel to create a chart or to compute some statistic.

11.4.2.12 Operational Support

Disco, Celonis, Perceptive, QPR, PPM, and Fujitsu can upload data periodically or incrementally. Analysis views are "refreshed" based on the new data. However, true predictive analyses, as described in Chap. 10, are seldom supported. QPR, PPM, and Rialto report integration efforts with dedicated prediction tools. However, these approaches do not seem to be process specific (i.e., the discovered model is not leveraged for prediction).

11.4.2.13 Scalability

Most of the commercial process mining tools have a good performance in terms of scalability and responsiveness. Some tools can even *handle event logs with billions of events, millions of cases, and hundreds of activities*. Loading such event logs may be time consuming (say up to an hour), but once the log is loaded analysis can be done within a few seconds.

Scalability depends on many different factors and not only the size of the event log. Some types of analysis are sensitive to the average trace length of cases, the number of distinct activities, or the number of attributes per event. Section 12.1.3 describes the key characteristics of logs relevant for scalability.

Organizations selecting a process mining tool are advised to test the scalability of tools on their own data using standard hardware. This is the only way to compare performance in a meaningful way (be aware of indexing and special hardware). See Chap. 12 for techniques to handle even larger event logs.

11.5 Outlook

It is impossible to give a complete overview of all products supporting process mining. Just ProM, the leading open-source process mining framework, already provides more than 1500 plug-ins. These plug-ins cover a wide range of analysis techniques. For example, all process discovery approaches described in this book are supported through ProM plug-ins. Moreover, ProM is not limited to process discovery and also supports conformance checking, social network analysis, bottleneck analysis, decision mining, operational support, verification, model conversion, etc. Most ProM plug-ins aim at use cases of *Type 1*. RapidProM, CoBeFra, and PMLAB support use cases of *Type 2*.

The 11 commercial process mining tools described in this chapter help to lower the threshold for process mining. Next to use cases of *Type 1*, also use cases of *Type 3* are supported using pre-configured dashboards and automated data extraction. Each of the eleven tools aims at supporting less experienced users. Sometimes process mining capabilities are embedded in larger software products. The scalability and usability of most commercial systems is good. Several tools can handle event logs with billions of events. However, compared to ProM there are also typical weaknesses such as the inability to discover concurrency well and the limited support for conformance checking. The focus is on performance analysis rather than conformance checking and precise models.

Since the process mining market is developing fast, readers are advised to test tools using their own event data. Even when tools look similar, differences in terms of practical usability and scalability may be significant.