Process Mining: Discovering and Improving Spaghetti and Lasagna Processes

Wil M.P. van der Aalst
Department of Mathematics and Computer Science, Eindhoven University of Technology - The Netherlands
Email: w.m.p.v.d.aalst@tue.nl

***Abstract*—Process mining is an emerging discipline providing comprehensive sets of tools to provide fact-based insights and to support process improvements. This new discipline builds on process model-driven approaches and data mining. This invited keynote paper demonstrates that process mining can be used to discover a wide range of processes ranging from structured processes (Lasagna processes) to unstructured processes (Spaghetti processes). For Lasagna processes, the discovered process is just the starting point for a broad repertoire of analysis techniques that support process improvement. For example, process mining can be used to detect and diagnose bottlenecks and deviations in (semi-)structured processes. The analysis of Spaghetti processes is more challenging. However, the potential benefits are substantial; just by inspecting the discovered model, important insights can be obtained. Process discovery can be used to understand variability and non-conformance. This paper presents the L^* life-cycle model consisting of five phases. The model describes how to apply process mining techniques.**

I. Introduction

Process mining, i.e., extracting valuable, process-related information from event logs, complements existing *Business Process Management* (BPM) approaches. BPM is the discipline that combines knowledge from information technology and knowledge from management sciences and applies this to operational business processes [1], [2]. It has received considerable attention in recent years due to its potential for significantly increasing productivity and saving cost. However, most BPM approaches use hand-made models as a starting point for analysis and enactment, i.e., factual event data about the process are not used systematically. The goal of process mining is to use event data to distill process related information, e.g., to automatically discover a process model by observing events recorded by some system or to check the conformance of a given model by comparing it with reality [3], [4].

Over the last decade, process mining techniques have matured. Today, it is possible to automatically extract process models from events logs, check the conformance of models by replaying logs, extend models with performance related information, use discovered models to predict flow times of running cases, etc. This paper does not aim to describe these techniques. Instead, we focus on the practical application of process mining. We first describe the L^* life-cycle model for process mining. This life-cycle model describes the various phases in a process mining project and is based on the practical application of process mining on more than 100 organizations.

The repeated application of process mining in various domains revealed that there is a *continuum* of processes ranging from *Lasagna processes* to *Spaghetti processes*. Often the terms "structured", "semi-structured", and "unstructured" are used to refer to this continuum. In a *structured process* (i.e., Lasagna process) all activities are repeatable and have a well defined input and output. In highly structured processes most activities can, in principle, be automated. In *semi-structured processes* the information requirements of activities are known and it is possible to sketch the procedures followed. However, some activities require human judgment and people can deviate depending on taste or the characteristics of the case being handled. In *unstructured processes* (i.e., Spaghetti process) it is difficult to define pre- and post-conditions for activities. These processes are driven by experience, intuition, trail-and-error, rules-of-thumb, and vague qualitative information.

After introducing the L^* life-cycle model for process mining, we discuss the characteristics of Lasagna and Spaghetti processes relevant for process mining. For Lasagna processes, the discovered process model is less relevant. However, the relationship between event logs and process models can be used to detect deviations, discover bottlenecks, suggest redesigns, predict delays, etc. For Spaghetti processes, the discovered model already provides important insights. Process mining tends to be challenging for Spaghetti processes. However, the potential benefits are substantial. Process automation and process improvement are only possible after understanding and streamlining such processes.

The paper concludes with some pointers to process mining software and the IEEE Task Force on Process Mining.

II. L^* Life-Cycle Model

Although there are many papers on process and data mining, few papers discuss how to apply process/data mining techniques to real-life processes. Some reference models describing the life-cycle of a typical data mining project have been proposed by academics and consortia of vendors and users. For example, the *CRISP-DM* (CRoss-Industry Standard Process for Data Mining) methodology identifies a life-cycle consisting of six phases: (a) business understanding, (b) data understanding, (c) data preparation, (d) modeling, (e) evaluation, and (f) deployment [5]. CRISP-DM was developed in the late nineties by a consortium driven by SPSS. Around the same period SAS proposed the *SEMMA* methodology consisting of five phases: (a) sample, (b) explore, (c) modify, (d) model,

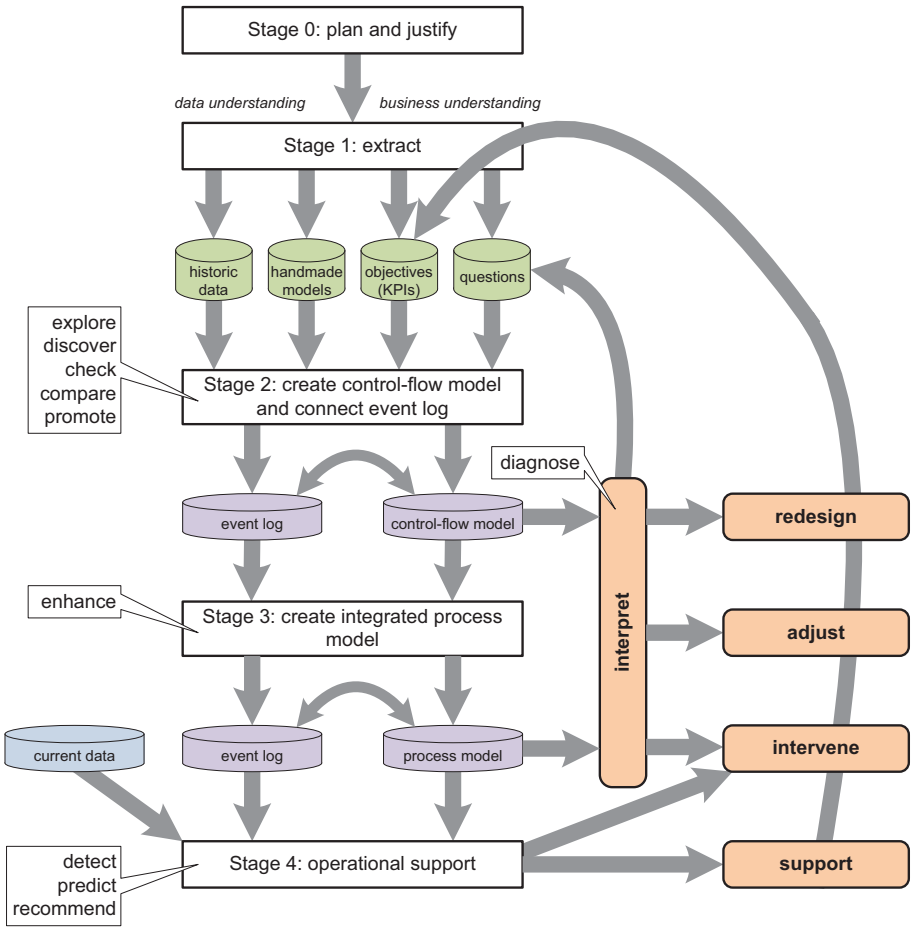

Fig. 1: The L^* life-cycle model describing a process mining project consisting of five stages: *plan and justify* (Stage 0), *extract* (Stage 1), *create control-flow model and connect event log* (Stage 2), *create integrated process model* (Stage 3), and *operational support* (Stage 4)

and (e) assess. Both methodologies are very high-level and provide little support. Moreover, existing methodologies are not tailored towards process mining projects. Therefore, we propose the L^* life-cycle model shown in Fig. 1. This five-stage model describes the life-cycle of a typical process mining project aiming to improve a Lasagna process.

In the remainder, we discuss each of the five stages shown in Fig. 1.

Stage 0: Plan and Justify

Any process mining project starts with a planning and a justification of the planned activities. Before spending efforts on process mining activities, one should anticipate benefits that may result from the project. There are basically three types of process mining projects:

- A *data-driven* (also referred to as "curiosity driven") process mining project is powered by the availability of event data. There is no concrete question or goal, however, some of the stakeholders expect that valuable insights will emerge by analyzing event data. Such a project has an explorative character.
- A *question-driven* process mining project aims to answer specific questions, e.g., "Why do cases handled by team X take longer than cases handled by team Y?" or "Why are there more deviations in weekends?".
- A *goal-driven* process mining project aspires to improve a process with respect to particular KPIs, e.g., cost reduction or improved response times.

For an organization without much process mining experience it is best to start with a question-driven project. Concrete questions help to scope the project and guide data extraction efforts.

Like any project, a process mining project needs to be planned carefully. For instance, activities need to be scheduled before starting the project, resources need to be allocated, milestones need to be defined, and progress needs to be monitored continuously.

Stage 1: Extract

After initiating the project, event data, models, objectives, and questions need to be extracted from systems, domain experts, and management.

Data extraction can be a time-consuming task. The challenge is not to the syntactical conversion of data; most efforts are related to finding the relevant data and to scope these data. For example, an SAP system has thousands of tables filled with data. Therefore, there is no such thing as extracting "the data" from SAP. Based on decisions made in Stage 0, specific data extractions are needed. Event logs need to satisfy two main requirements: (a) events need to be ordered in time and (b) events need to be correlated (i.e., each event needs to refer to a particular case).

As Fig. 1 shows, it is possible that there are already handmade (process) models. These models may be of low quality and have little to do with reality. Nevertheless, it is good to collect all models present and exploit existing knowledge as much as possible. For example, existing models can help in scoping the process and judging the completeness of event logs.

In a goal-driven process mining project, the objectives are also formulated in Stage 1 of the L^* life-cycle. These objectives are expressed in terms of KPIs. In a question-driven process mining project, questions need to be generated in Stage 1. Both questions and objectives are gathered through interviews with stakeholders (e.g., domain experts, end users, customers, and management).

Stage 2: Create Control-Flow Model and Connect Event Log

Control-flow forms the backbone of any process model. Therefore, Stage 2 of the L^* life-cycle aims to determine the de facto control-flow model of the process that is analyzed. The process model may be discovered using process discovery techniques such as the α-algorithm, heuristic mining, fuzzy mining, and genetic mining (activity *discover* in Fig. 1). However, if there is a good process model present, it may be verified using conformance checking (activity *check*) or judged against the discovered model (activity *compare*). It is even possible to merge the handmade model and the discovered model (activity *promote*). After completing Stage 2 there is a control-flow model tightly connected to the event log, i.e., events in the event log refer to activities in the model. *This connection is crucial for subsequent steps.* If the fitness of the model and log is low (say below 0.8), then it is difficult to move to Stage 3. However, by definition, this should not be a problem for a Lasagna process.

The output of Stage 2 may be used to answer questions, take actions, or to move to Stage 3. As Fig. 1 shows, the output (control-flow model connected to an event log) needs to be interpreted before it can be used to answer questions or trigger a redesign, an adjustment, or an intervention.

Stage 3: Create Integrated Process Model

In Stage 3, the model is enhanced by adding additional perspectives to the control-flow model (e.g., the organizational perspective, the case perspective, and the time perspective). The connection between events in the log and activities in the process model can be used to merge the different perspectives into a single model, e.g., the timestamps in the log can be

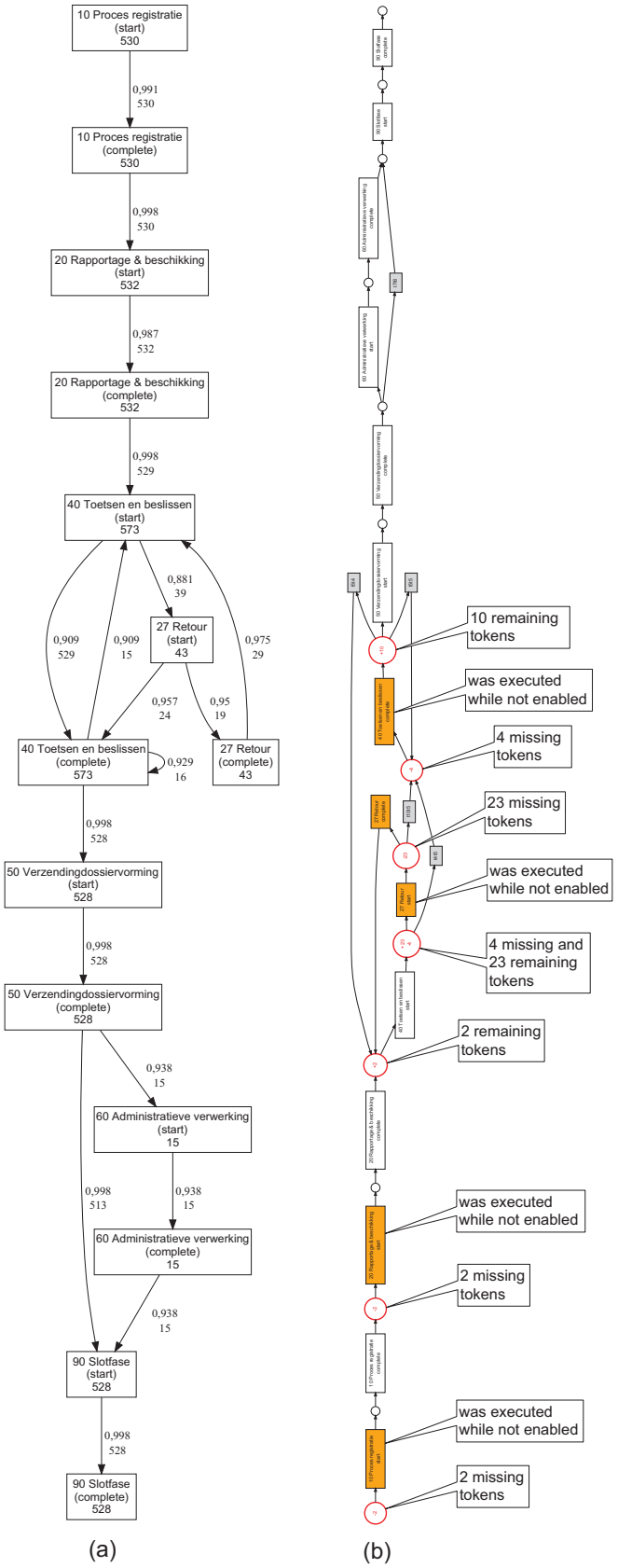

Fig. 2: The C-net discovered using the heuristic miner (a) and the corresponding Petri net with missing and remaining tokens after replay (b).

used to associate durations to activities and information about resources can be used to discover resource allocation rules. The result is an integrated process model that can be used for various purposes. The model can be inspected directly to better understand the as-is process or to identify bottlenecks. Moreover, a complete process model can also be used as the starting point for simulations [6].

The output of Stage 3 can also be used to answer selected questions and take appropriate actions (redesign, adjust, or intervene). Moreover, the integrated process model is also input for Stage 4.

Stage 4: Operational Support

Stage 4 of the L^* life-cycle is concerned with three operational support activities: *detect*, *predict*, and *recommend*. For instance, it is possible to predict the remaining flow time for running cases or to recommend suitable actions based on historic information. As shown in Fig. 1, Stage 4 requires current data ("pre mortem" data on running cases) as input. Moreover, the output does not need to be interpreted by the process mining analyst and can be directly offered to end users. For example, a deviation may result in an automatically generated e-mail sent to the responsible manager. Recommendations and predictions are presented to the persons working on the corresponding cases.

Note that operational support is the *most ambitious* form of process mining. This is only possible for Lasagna processes. Moreover, there needs to be an advanced IT infrastructure that provides high-quality event logs and allows for the embedding of an operational support system.

III. Lasagna Processes

Unlike Spaghetti processes, Lasagna processes have a clear structure and most cases are handled in a prearranged manner. There are relatively few exceptions and stakeholders have a reasonable understanding of the flow of work. It is impossible to define a formal requirement characterizing Lasagna processes. As a rule of thumb we use the following informal criterion: *a process is a Lasagna process if with limited efforts it is possible to create an agreed-upon process model that has a fitness of at least 0.8*, i.e., more than 80% of the events happen as planned and stakeholders confirm the validity of the model. This implies (assuming that a suitable event log can be extracted) that, in principle, all stages of the L^* life-cycle can be executed.

Figure 2 shows an example of a Lasagna process discovered for one of the so-called WMO processes of a Dutch municipality. WMO (Wet Maatschappelijke Ondersteuning) refers to the social support act that came into force in The Netherlands on January 1st, 2007. The aim of this act is to assist people with disabilities and impairments. The WMO act forced all Dutch municipalities to implement various supporting processes. Figure 2 is based on the WMO process for handling requests for household help. In a period of about one year, 528 requests for household WMO support were received. These 528 requests generated 5498 events. Figure 2(a) shows a so-called C-net discovered using the heuristic miner [3], [7]. The numbers generated by the heuristic miner show the flow of tokens, e.g., activity "10 Process registratie" was executed 530 times. The C-net was translated into an equivalent Petri net with silent transitions as shown in Fig. 2(b). The fitness of the discovered process is 0.99521667 [3], [8]. This implies that almost all behavior captured in the event log can be reproduced by the discovered process model. Of the 528 cases, 496 cases fit perfectly (i.e., can be replayed from begin to end) whereas for 32 cases there are missing or remaining tokens. The missing and remaining tokens show where the model and log deviate. For example, for two cases the activity "40 toetsen en beslissen" (evaluate and decide) was not started although it should have. Activity "20 Rapportage & beschikking" (report and intermediate decision) was started twice while this was not possible according to the model. Figure 2(b) illustrates that process mining can be used to measure conformance and diagnose deviations.

Most conformance checking techniques are based on replaying the event log on the process model. Since in most logs events have timestamps, the same mechanism can be used to analyze time-related aspects [3]. Replay can be used to discover and diagnose bottlenecks. Figure 3 shows some results for the WMO process for handling requests for household help. The different parts of the process model can be colored to indicate waiting and service times. Figure 3 shows that it is also possible to point at two arbitrary points in the process (say X and Y) and measure the number of cases that flow from X to Y, the average time it takes to flow from X to Y, and all kinds of other statistics (variance, minimum, etc.). Hence, after creating the control-flow model and connecting the event log to the model (Stage 2 in Fig. 1), it is possible to create an integrated process model also incorporating performance related information. The integrated process model may also contain information about data, decision rules, resources, roles, organizational units, etc. For Lasagna processes these can be merged into a single model (Stage 3 in Fig. 1). In fact, such a model can be used for operational support (Stage 4 in Fig. 1). For example, it is possible to predict the remaining flow time of a WMO application or to recommend activities or resources to minimize flow time. Conformance checking can also be done on the fly, i.e., it is possible to generate a warning the moment a deviation occurs.

For Lasagna processes it is, in principle, possible to execute all phases of the L^* life-cycle model. As shown in Fig. 1, the results can be used to redesign such processes (e.g., to minimize costs or flow times), to adjust processes to changing circumstances, to intervene (e.g., to reallocate a worker due to an unusual percentage of deviations), and to provide support (e.g., predicting flow times).

IV. Spaghetti Processes

Spaghetti processes are the counterpart of Lasagna processes. Because Spaghetti processes are less structured, only a subset of available process mining techniques is applicable.

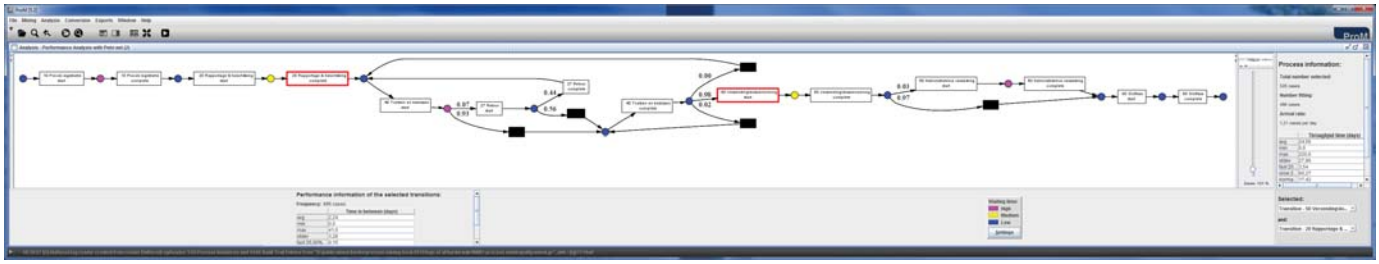

Fig. 3: Screenshot of ProM 5.2 while analyzing the bottlenecks in the process. The mean flow time of fitting cases is 24.66 days. Most time is spent on the activities “10 Process registratie”, “40 Toetsen en beslissen”, and “60 Administratieve verwerking”. The average time in-between the completion of activity “10 Rapportage & beschikking” and “50 Verzending/dossiervorming” is 2.24 days

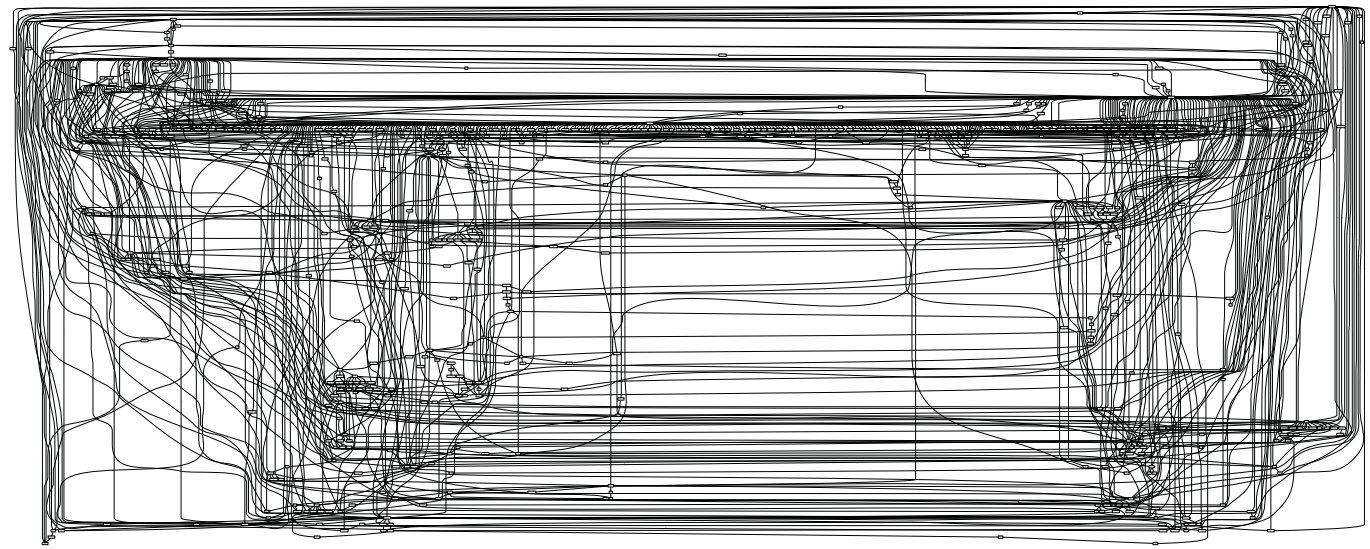

Fig. 4: Spaghetti process describing the diagnosis and treatment of 2765 patients in a Dutch hospital. The process model was constructed based on an event log containing 114,592 events. There are 619 different activities (taking event types into account) executed by 266 different individuals (doctors, nurses, etc.)

For instance, it makes no sense to aim at operational support activities if there is too much variability.

Figure 4 illustrates why unstructured processes are called Spaghetti processes. The model is based on event data related to 2765 patients in a Dutch hospital. The process model depicted was obtained using the heuristic miner with default settings. Hence, low frequent behavior has been filtered out. Nevertheless, the model is too difficult to comprehend. Note that this is not necessarily a problem of the discovery algorithm. Activities are only connected if they frequently followed one another in the event log. Hence, the complexity shown in Fig. 4 reflects reality and is not caused by the discovery algorithm.

Clearly, only the initial stages of the L^* life-cycle model are applicable for Spaghetti processes such as the process shown in Fig. 4. To enable history-based predictions and recommendations it is essential to first make the “Spaghetti-like” process more “Lasagna-like”. In fact, Stage 3 and Stage 4 will be too ambitious for most Spaghetti processes. It is always possible to generate process models as illustrated by Fig. 4. Moreover, it is often also possible to create models for other perspectives, e.g., flow times, social networks, and decision models. However, it is very unlikely that all of these can be folded into a meaningful comprehensive process model as the basis (the control-flow discovered) is too weak.

Spaghetti processes are more difficult to analyze than Lasagna processes. Nevertheless, such processes are very interesting from the viewpoint of process mining as they often allow for various improvements. A highly-structured well-organized process is often less interesting in this respect; it is easy to apply process mining techniques but there is also little improvement potential. Therefore, one should not shy away from Spaghetti processes as these are often appealing from a process management perspective.

Process discovery is a challenging task. Event logs are typically far from complete, i.e., often only a fraction of the possible behavior is captured in the log. Moreover, event logs do not contain negative examples, i.e., only positive example behavior is given. The fact that something does not happen in an event log does not mean that it cannot happen. Process

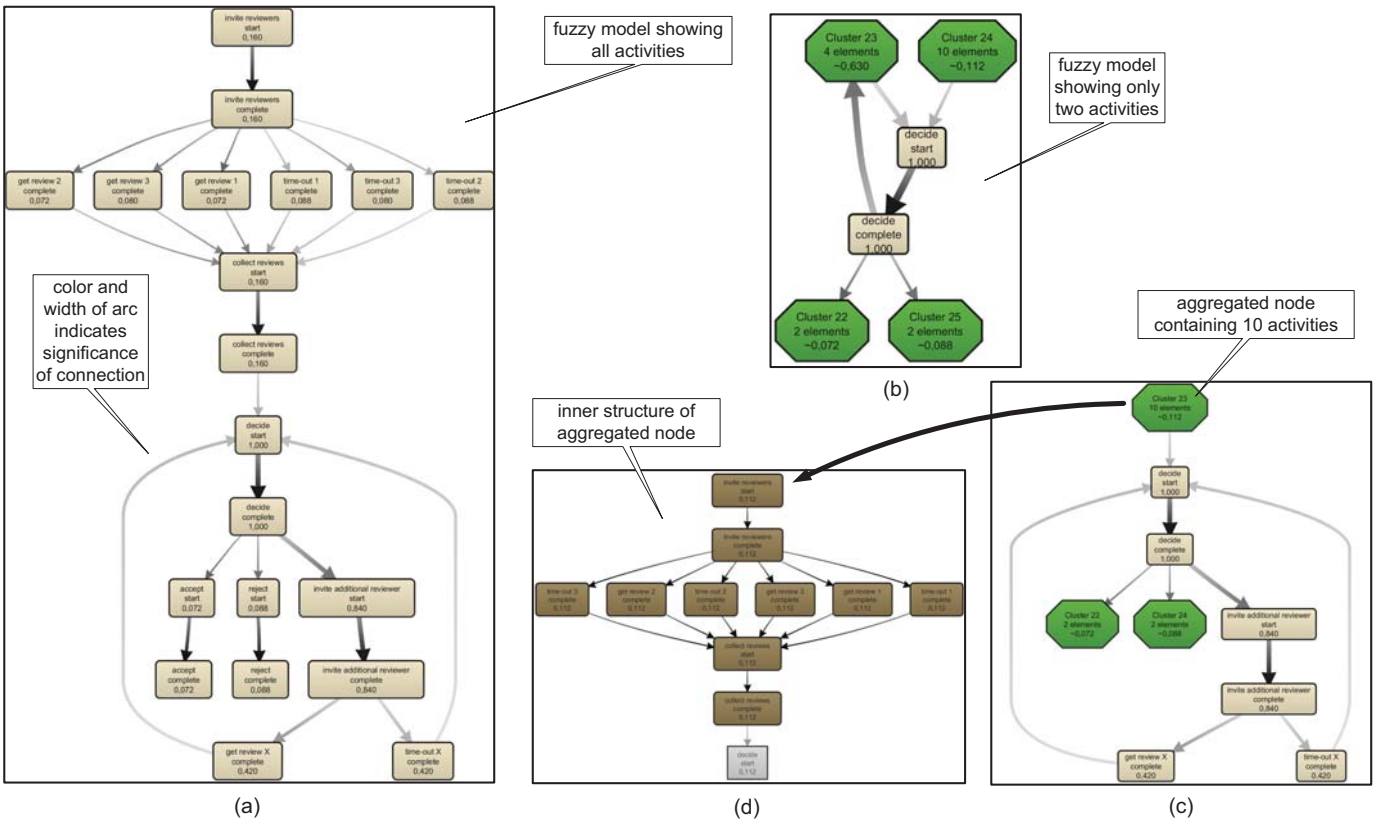

Fig. 5: Three business process maps obtained using ProM's Fuzzy Miner. The most detailed fuzzy model (a) shows all activities. The least detailed fuzzy model (b) shows only two activities; all other activities are aggregated into so-called "cluster nodes". The third fuzzy model (c) shows six activities. For one of the aggregate nodes, the inner structure is shown (d)

discovery techniques need to balance four criteria [3]: *fitness* (the discovered model should allow for the behavior seen in the event log), *precision* (the discovered model should not allow for behavior completely unrelated to what was seen in the event log), *generalization* (the discovered model should generalize the example behavior seen in the event log), and *simplicity* (the discovered model should be as simple as possible).

The Fuzzy Miner of ProM [9] aims to balance these four criteria. Figure 5 illustrates this approach. Unlike classical techniques, the Fuzzy Miner allows for seamlessly zooming in and out (as is shown in Fig. 5). The three fuzzy models shown in Fig. 5 are all based on the same event log. Figure 5(a) shows the most detailed view. All activities are included. The color and width of the connections indicate their significance. Figure 5(b) shows the most abstract view. Figure 5(c) shows a model generated using intermediate settings. The top-level model shows the six most frequent activities. The other activities can be found in the three cluster nodes. Figure 5(d) shows the inner structure of one of the cluster nodes in Fig. 5(c).

When zooming out using Google maps, less significant elements are either left out or dynamically clustered into aggregate shapes. For example, streets and suburbs amalgamate into cities. This is similar to the zoom functionality provided by ProM's Fuzzy Miner as was illustrated using Fig. 5. See [3] for more information on state-of-the-art process discovery approaches and open challenges.

V. Tool Support

Many vendors offer *Business Intelligence* (BI) software products. Some of the most widely used BI products are IBM Cognos Business Intelligence (IBM), Oracle Business Intelligence (Oracle), and SAP BusinessObjects (SAP). Unfortunately, most of these products are data-centric and focus on rather simplistic forms of analysis. Data mining tools provide more advanced forms of analysis. However. also these systems are typically data centric, focusing on classification (e.g., decision trees), regression, clustering, and association rules.

Process mining research started in the late nineties. Initially, researchers developed simple prototypes (MiMo, EMiT, InWolvE, Process Miner) restricted to control-flow discovery [10]. An important innovation was 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 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 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 [8]. *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.

ProM 6 (released in November 2010) is based on XES rather than MXML. XES is the new 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 [11]. 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. 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.

The functionality of ProM is unprecedented, i.e., there is no product offering a comparable set of process mining algorithms. However, the tool 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, there is a growing number of commercially available software products offering process mining capabilities. Some of these products embed process mining functionality in a larger system, e.g., Pallas Athena embeds process mining in their BPM suite BPM|one (www.pallas-athena.com). Other products aim at simplifying process mining using an intuitive user interface, e.g., *Reflect* by Futura Process Intelligence (www.futuratech.nl). As mentioned before, the large number of plug-ins of ProM can be rather overwhelming. Other examples of commercial products supporting process mining are *ARIS Process Performance Manager* (Software AG), *Enterprise Visualization Suite* (Businesscape), *Interstage BPME* (Fujitsu), *Process Discovery Focus* (Iontas), and *ProcessAnalyzer* (QPR). Besides these commercial initiatives, there are also several research groups developing stand-alone process discovery tools.

VI. IEEE Task Force on Process Mining

More and more people, both in industry and academia, consider process mining as one of the most important innovations in the BPM field. Process mining joins ideas of process modeling and analysis on the one hand and data mining and machine learning on the other. Therefore, the IEEE established a *Task Force on Process Mining* in the context of the Data Mining Technical Committee (DMTC) of the Computational Intelligence Society (CIS). The goal of this task force is to promote the research, development, education and understanding of process mining. See http://www.win.tue.nl/ieeetfpm/ for more information about the IEEE Task Force on Process Mining. The reader is encouraged to start using existing process mining techniques and tools, and contribute to the growing body of knowledge.

Acknowledgment

The authors would like to thank the members of the IEEE Task Force on Process Mining (www.win.tue.nl/ieeetfpm/) and all that contributed to the development of ProM (www.processmining.org).

References

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

[2] M. Weske, *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag, Berlin, 2007.

[3] W. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.

[4] W. van der Aalst, H. Reijers, A. Weijters, B. van Dongen, A. Medeiros, M. Song, and H. Verbeek, "Business Process Mining: An Industrial Application," *Information Systems*, vol. 32, no. 5, pp. 713–732, 2007.

[5] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "CRISP-DM 1.0: Step-by-step data mining guide," 2000, www.crisp-dm.org.

[6] W. van der Aalst, "Business Process Simulation Revisited," in *Enterprise and Organizational Modeling and Simulation*, ser. Lecture Notes in Business Information Processing, J. Barjis, Ed., vol. 63. Springer-Verlag, Berlin, 2010, pp. 1–14.

[7] A. Weijters and W. van der Aalst, "Rediscovering Workflow Models from Event-Based Data using Little Thumb," *Integrated Computer-Aided Engineering*, vol. 10, no. 2, pp. 151–162, 2003.

[8] A. Rozinat and W. van der Aalst, "Conformance Checking of Processes Based on Monitoring Real Behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, 2008.

[9] C. Günther and W. van der Aalst, "Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics," in *International Conference on Business Process Management (BPM 2007)*, ser. Lecture Notes in Computer Science, G. Alonso, P. Dadam, and M. Rosemann, Eds., vol. 4714. Springer-Verlag, Berlin, 2007, pp. 328–343.

[10] W. van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters, "Workflow Mining: A Survey of Issues and Approaches," *Data and Knowledge Engineering*, vol. 47, no. 2, pp. 237–267, 2003.

[11] C. Günther, "XES Standard Definition," www.xes-standard.org, 2009.