Christian Czarnecki et al. (Hrsg.): Workshop der INFORMATIK 2018,
Lecture Notes in Informatics (LNI), Gesellschaft fr Informatik, Bonn 2018 239

Folding Marked Generalized Stochastic Petri Nets for Time Prediction in Business Processes

Stephan A. Fahrenkrog-Petersen [1], Matthias Weidlich[2]

Abstract: Generalized Stochastic Petri Nets (GSPNs) can be used for performance analysis of business processes. Recently, it was shown that foldings of a GSPN, i.e., a set of model reduction rules, help to avoid over-fitting of the model with respect to the performance characteristics of a process. Yet, these foldings ignore the marking of a GSPN and, thus, are applicable solely for steady-state analysis. In this paper, we discuss how foldings may be lifted to marked nets and provide an assessment of stateful foldings for sequential GSPNs.

Keywords: Business Process Management; Process Mining; Generalized Stochastic Petri nets; Model Simplification

1 Introduction

The field of process mining is devoted to the analysis of business processes based on event logs [vdA16], which are recorded during process execution. Specifically, process mining includes methods to enrich business process models with performance information. For instance, Petri-net models constructed from an event log may be enriched with performance information to obtained Generalized Stochastic Petri Nets (GSPNs), which add capacities, durations, and transition weights to classic Petri-nets. Based thereon, questions related to the sojourn time of a process may be answered, e.g., how long the treatment of a patient in a hospital will take on average [OTB11].

Recently, it was argued that GSPNs mined from event logs shall be simplified to avoid overfitting of the model in the performance dimension [Se18]. In this context, overfitting refers to annotations of a model that simply represent a specific dataset, i.e., that map the performance characteristics as they are directly observed in an event log. As such, an overfitting model does not capture the general performance characteristics of the considered process. To achieve generalisation of the exemplary performance characteristics observed in the log, a set of foldings has been proposed, i.e., structural transformations of GSPNs that reduce the model and aggregate the performance annotations. However, these foldings only

[1] Humboldt-Universität zu Berlin, Informatik, Rudower Chaussee 25, 12489 Berlin, Germany, fapeters@hu-berlin.de
[2] Humboldt-Universität zu Berlin, Informatik, Rudower Chaussee 25, 12489 Berlin, Germany, matthias.weidlich@hu-berlin.de

considered unmarked GSPNs and, thus, are restricted to steady-state analysis of processes that ignores the current state of the process. In state-based analysis, in turn, we are interested in model-based predictions at a certain point of time, so that the current state of the process needs to be incorporated. For instance, instead of the treatment time of patients on average (steady-state analysis), we are interested in how long it will take to provide treatment for a specific patient at the current point in time (state-based analysis). Such predictions are needed, for example, when deciding on where to route patients.

In order to incorporate the state of the process in prediction, analysis needs to consider the marking of a GSPN. In this paper, we argue that existing foldings of GSPNs may be lifted from unmarked GSPNs to marked GSPNs, transforming not only the structure, but also the marking of a GSPN.

Below, we exemplify this idea for a folding of a sequential GSPN. However, we argue that this idea may also be applied to other types of foldings, i.e., those for parallel, exclusive, and repetitive process structures.

2 Preliminaries

GSPNs extend traditional Petri-Nets [Ma95], modelling transitions that take time to fire. In GSPNs, there exist two types of transitions: *immediate transitions* and *timed transitions*. *Immediate transitions* fire without consuming time, but have weights that become relevant whenever two immediate transitions compete for the same token. Weights determine the probabilities for firing one of these transitions. *Timed transitions* have durations, thereby modelling tasks that take time. These transitions are assigned a capacity, which represents how often a transition can be enabled concurrently. Following [Se18], we define GSPNs:

Definition 1 *A GSPN is a tuple* $G = (P, T, F, \gamma, \delta, \omega)$ *where:* P *is the set of places,* $T = T_i \cup T_t$ *is the set of transitions, which are immediate* (T_i) *or timed* (T_t)*, respectively,* $F \subseteq (P \times T) \cup (T \times P)$ *is the flow relation,* $\gamma : T_t \longrightarrow \mathbb{R}_0^+$ *assigns capacities to timed transitions (work units per time unit),* $\delta : T_t \longrightarrow \mathbb{R}_0^+$ *assigns expected duration's to timed transitions,* $\omega : T_t \longrightarrow [0, 1]$ *assigns weights to immediate transitions.*

Semantics of a GSPN are defined as a token game: A marking $M : P \longrightarrow \mathbb{N}_0$ assigns tokens to places. A transition can be *fired*, if it is *enabled*. A transition is enabled, if all places in its preset are marked. An immediate transition will be fired immediately after it is enabled. In contrast, a timed transition will start a clock, once it is enabled. The transition will be fired if the time of the clock has passed. The rate λ of a transition is given as ratio of its capacity and its duration, i.e., $\lambda(t) = \frac{\gamma(t)}{\delta(t)}$ for some $t \in T_t$.

Additionally we use the following notations: M_0 is the empty marking, with $M_0(p) = 0$ for all $p \in P$. We consider M_0 as the initial marking of a GSPN. Given a marking M, the set of *reachable markings*, denoted by $R(M)$, contains all markings that are obtained by firing a sequence of transitions, starting in marking M.

3 Foldings of Marked GSPNs

A folding, as introduced in [Se18], takes as input a GSPN and returns a reduced GSPN. As such, foldings are applicable for particular structures of GSPNs. A simple folding, illustrated in Figure 1, is the *sequence folding*. It folds several timed transitions that are ordered sequentially into a single timed transition. In Figure 1, a sequence of three timed transitions is folded into the transition labelled with τ. In [Se18], it was shown how the capacity and duration of this transition are obtained: It is assigned the sum of capacities and the sum of durations of the original transitions.

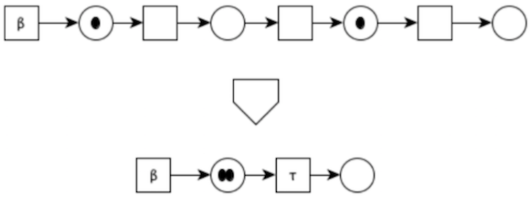

Fig. 1: Example of a sequence folding that incorporates a marking.

If a GSPN is marked, applying a folding requires the definition of a marking for the folded net, based on the marking of the original net. Put differently, the tokens of the marking in the original need to be distributed among the places of the folded net. Below, we describe our approach to obtain such a distribution.

Foldings yield a net that contains a single timed transition, with one place in its preset and one place in its postset. Hence, tokens may only be distributed among these two places in the folded net. This means that the *estimated sojourn time* of the generated timed transition τ, denoted by ES_τ, is calculated as follows:

$$ES_\tau = \frac{M_b}{\lambda(\tau)} \tag{1}$$

where $\lambda(\tau) = \frac{\gamma(\tau)}{\delta(\tau)}$ is the rate of transition τ and M_b is the number of tokens in the place in the preset of τ. The number of tokens for the place in the postset of τ, henceforth referred to as M_a, is then derived by taking the remainder of the tokens that need to be distributed:

$$M_a = M_{orig} - M_b \tag{2}$$

with M_{orig} is the total number of tokens that needs to be distributed in the folded net. For most types of foldings, i.e., those related to sequential, exclusive, and repetitive structures, this number is equal to the total number of tokens in the original net. An exception is the *AND-folding* which folds a number of concurrently enabled timed transitions into a single one. In that case, M_{orig} amounts to the number of tokens in the original subnet, divided by the number of potentially concurrently enabled transitions in the original net. The reason is that the sojourn time over these concurrently enabled transitions is determined by the sojourn time of one of the respective tokens (intuitively, the 'slowest' one).

Based on the above, we suggest a local minimum approach to compute the number of tokens M_b that should be put in the place in the preset of transition τ in the folded net. Computing M_b is sufficient since the number of tokens M_a in the place in the postset of transition τ is then derived based on M_b and M_{orig}. Specifically, we compute M_b by solving the following minimization problem:

$$\min(f(M_b) \mid M_b \in \mathbb{N}, 0 \leq M_b \leq M_{orig}) \quad \text{with} \quad f(M_b) = \left| ES_{orig} - \frac{M_b}{\lambda(\tau)} \right| \tag{3}$$

where ES_{orig} is the estimated sojourn time of the original net. In other words, we find the optimal number of tokens to put into the place in the preset of transition τ, so that the difference in sojourn times of the original and the folded net is minimal.

An important property of foldings is properness [Se18]. That is, a folding shall preserve stability of a GSPN: If a net has a finite expected waiting time value, the folded net should show the same property. In GSPN terms, this is captured as follows. Given a GSPN $G = (P, T, F, \gamma, \delta, \omega)$, a transition $t \in T_t$ is stable, if the marking $M_h(p) = 0$, for all $p \in \{p \in P \mid (p, t) \in F\}$ is a home marking for the initial marking M_0, i.e., for all $M' \in R(M_0)$ it holds that $M_h \in R(M')$.

The foldings for unmarked GSPNs introduced in [Se18], including the sequence folding illustrated above, have been shown to be proper. They preserve the stability of a GSPN. We now show for the sequence folding that properness also holds if the marking is considered in the folding, following the above approach to distribute the tokens.

Theorem 1. *If a marked GSPN is stable, then the GSPN obtained by the sequence folding that distributes the tokens of the marking is also stable.*

Proof. In [Se18], the properness of the sequence folding was shown for unmarked GSPNs. It is known that a GSPN is only stable, if the rates of all timed transitions $t \in T_t$ are larger than the rate of arrivals into the net, i.e., $\beta < \lambda_t$ [Br08]. In a GSPN that comprises a sequence of timed transitions, the rate of arrivals into the net, i.e., β, is also the rate of arrivals into each of the timed transitions [Ja04]. In a folded, marked GSPN, the arrival rate increases to $\beta + x_t$, where $x_t \geq 0$ for every transition t that follows a folded transition. The increase of the rate is the result of the redistribution of tokens: The folded net comprises less places compared to the original net. Using the above approach for the distribution of tokens, the number of tokens after a sequence folding remains the same. Hence, the folded net must contain a least one place with a higher number of tokens than in the marking of the original net. Transitions in the postset of this place can thus fire with a rate larger than β, since $\beta < \lambda_t \in T_t$ still holds true [Se18]. This is the reason why we need to introduce x_t as the one-time impact of the folding of the marking. The size of x increases with the number of tokens involved in the folding. However, x decreases over time, meaning that the impact of the redistributed tokens on the rate decreases as more and more tokens over times are consumed by the net. Hence, at some point, it holds that $\beta + x_t < \lambda_t$, which is a sufficient condition for the stability of the GSPN.

4 Assessment

Theoretical considerations. Comparing foldings for unmarked and marked GSPNs, we observe the following: Either type of folding accelerates the GSPN. In both folding variants, the time of a single token passing through all timed transitions is equivalent, if each enabled timed transition can fire immediately. When considering the marking in the folding, however, this assumption may not hold true. As such, the acceleration of the GSPN is potentially larger if the marking is considered. As a consequence, foldings that incorporate markings will always be at least as accurate than foldings that neglect the marking.

Experimental assessment. We further conducted an experiment to assess the impact of considering the marking of a GSPN in performance analysis of a business process, and also explore the influence of considering the marking in the folding.

The experiment has been conducted using the dataset and methodology of [Se18]. That is, we used data of a treatment process of a large outpatient hospital to discover a GSPN and enrich it with performance information. Due to our focus on the sequence folding, we considered a projection of the event log on a sequence of major treatment steps. We filtered all traces of the event log that do not contain the respective sequence. This led to a sample of 1,549 traces, out of a total of 4,282 traces in the event log. Furthermore, for analysis based on stateful, i.e., marked GSPNs, the state is derived solely for the treatment steps that are part of the selected sequence. Extraction of performance information was done based on the duration of treatment steps as recorded in the event log. Moreover, we approximated the capacity per treatment step from the event log, by considering the maximal number of concurrent executions of a treatment step at a single point in time.

We then focused on the *estimated sojourn time* of the process. We computed the root-mean squared error (RMSE), a standard statistical accuracy measure, by comparing the model-based prediction with the observed cycle time (OCT) in the event log. Specifically, we consider the following experimental scenarios:

(1) OCT vs. prediction based on unfolded, stateless model
(2) OCT vs. prediction based on unfolded, stateful model
(3) OCT vs. prediction based on folded, stateful model

The obtained results are shown in Figure 2. We first notice a significant improvement of the prediction once the state of the model is considered. Moving from a stateless prediction to a stateful one significantly lowers the RMSE. This is expected as the effects that are due to temporary congestion are now incorporated. Folding the stateful model, however, increases the RMSE. While such an increase is, in general, expected, we observe a drastic increase in our particular scenario. Further work is needed to determine the exact reason for this observation. A hypothesis is that the aforementioned estimation of capacities for the treatment steps in the process is too simplistic and, hence, introduces a strong bias.

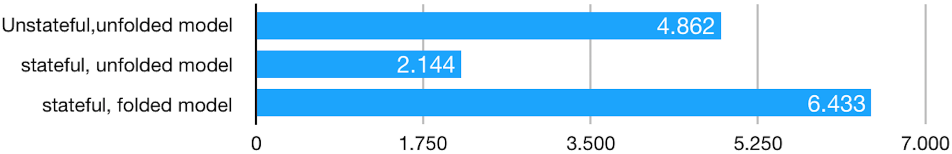

Fig. 2: Results of the experimental assessment, with the RMSE being reported in seconds.

5 Conclusion

In this work, we took up recent ideas on performance analysis of business processes that are grounded in GSPNs. Existing approaches to avoid overfitting of GSPNs in the performance dimension are applicable only for unmarked nets. While such stateless models are useful for steady-state analysis, they are not suited for predictions made at a particular point in time that shall incorporate the current state of the process. We therefore discussed how foldings can be lifted from unmarked GSPNs to marked GSPNs, by distributing the tokens of the marking of the original net in the folded one. Moreover, we discussed the properness of the sequence folding once the marking is incorporated.

References

[Br08] Bramson, Maury: Stability of queueing networks. 2008.

[Ja04] Jackson, J. R: Jobshop-like queueing systems. Management science, 50(12_supplement):1796–1802, 2004.

[Ma95] Marsan, M. Ajmone; Balbo, G.; Conte, G.; Donatelli, S.; Franceschinis, G.: Modelling with Generalized Stochastic Petri Nets. John Wiley and Sons, 1995.

[OTB11] Olanrewaju, S. A.; Takayesu, J. K.; Bohan, S. J.: Framework for analyzing wait times and other factors that impact patient satisfaction in the emergency department. Journal of Emergency Medicine, 41(6):686–692, 2011.

[Se18] Senderovich, A.; Shleyfman, A.; Weidlich, M.; Gal, A.; Mandelbaum, A.: To aggregate or to eliminate? Optimal model simplification for improved process performance prediction. Information Systems, 2018.

[vdA16] van der Aalst, W. M. P.: Process Mining - Data Science in Action, Second Edition. Springer, 2016.

Desktop Activity Mining - A new level of detail in mining business processes

Christian Linn[1], Phileas Zimmermann[2], Dirk Werth[3]

Abstract:

New analysis and automation technologies are significantly changing the way how business process management is performed. Especially Robotic Process Automation (RPA) is rapidly gaining importance as a method to automate office processes. An efficient automation of office processes however requires detailed information about all user activities related to the process. While process mining techniques can in principle be used to discover processes in a data-driven way, the existing approaches are not able to gather information in a level of detail required for automation purposes. That is why in particular the configuration of RPA systems is a labor and knowledge-intensive task that is based on a human expert, modeling all process variations in detail. In this paper, we present Desktop Activity Mining as a new approach to mine detailed process activity data. The concept is to record the detailed desktop activities of all users performing an office process and consolidate the process variations with process mining techniques to discover an integrated process model. As a proof of concept, we realized a prototypical implementation. Our findings suggest that Desktop Activity Mining holds the potential to optimize not only process automation but also to derive a new level of detail in mining and analyzing business processes.

Keywords: Business Process Management; Robotic Process Automation; Process Mining; Desktop Mining.

1 Introduction / Motivation

In today's competitive environment, it becomes increasingly important for companies to understand and optimize their internal business processes. Classical business process management relies on abstract process models which are used to define, plan and control the execution of a process [VDA03]. In recent years new technologies for analyzing and automating business processes emerged that are significantly changing the way how business process management is performed.

The development of process mining techniques introduces a data-driven method to discover, verify or improve process models [VDA12]. It relies on transactional data from information

[1] August-Wilhelm Scheer Institute, Uni-Campus D 5 1, 66123 Saarbrücken, Germany, christian.linn@aws-institut.de

[2] August-Wilhelm Scheer Institute, Uni-Campus D 5 1, 66123 Saarbrücken, Germany, phileas.zimmermann@aws-institut.de

[3] August-Wilhelm Scheer Institute, Uni-Campus D 5 1, 66123 Saarbrücken, Germany, dirk.werth@aws-institut.de

systems that contain information about the executed activities. Besides analysis and monitoring, also the automation of business processes profits from new technologies. Especially the automation of manufacturing processes has significantly increased in the last years, mainly driven by the technological progress in robotics [HNP08]. To automate office processes, Robotic Process Automation (RPA), the approach of automating software tasks by utilizing software robots, is gaining cross-industry wide recognition [LWC15] [WLC15]. The advantage of RPA is that it operates on the User Interface (UI) layer, i.e. mimics the interaction of a human with a computer, by e.g. performing mouse-clicks or keyboard input [LW]. Contrary to classical IT automation approaches, a complex integration of different IT systems is not necessary. As a consequence RPA offers companies a high return-on-investment when automating internal office processes [LWC15].

But there are also challenges which arise with this new automation approach: The realization of a reliable automation requires detailed information about the user activities related to the process. In order to configure an RPA system, i.e. defining the execution path, the process needs to be modeled with a level of detail that is sufficient to execute the process without further manual input. Currently, this needs to be done manually, by a person with good process modeling and domain knowledge, defining the activities the software robot has to perform. Thus the configuration process is a highly time-intensive task, especially in cases in which different execution paths and exceptions need to be considered. Hence, an approach is needed that is able to automatically capture detailed information about the variations of a given process. Process Mining could be a way to solve this problem as it relies on transactional data that is logged in the software systems of a company. However, a big disadvantage of this concept is that in reality, many activities are not executed within these information systems alone and hence are not fully logged and documented. A simple example is writing an email or performing searches in web browsers. Furthermore, many companies still rely on IT legacy systems that do not log any sort of data [BG11]. Therefore, a classical process mining approach does not have access to the activities performed in such systems.

In the following paper, we introduce the new concept of Desktop Activity Mining (DAM) which is designed to solve the above-mentioned issues by introducing a new level of detail in process activity mining. The aim is to a) capture user actions that are not logged by information systems, b) obtain a complete set of process variations and c) derive a process model and documentation on a maximum level of detail.

The research in this paper follows a design science approach as defined by [Vo04]. Following the methodology, we first show relevance and rigor in our research design by giving insights into Process Automation, Process Mining and related research in Section 2. By discussing requirements and shortcomings of mentioned research fields, we motivate the research in this paper. Grounded in this discussion we present the concept of Desktop Activity Mining in Section 3, which is the novel artifact. Furthermore, we demonstrate the feasibility of our concept by discussing a proof of concept in Section 4. Finally we conclude in Section 5 with an outlook on future work.

2 Related Work

Business Process Management (BPM) is concerned with the discovery, automation, optimization, analysis and modeling of business processes [JN14]. BPM encompasses a wide array of sub-fields, such as Process Mining (PM) and Business Process Automation (BPA). In recent years new technologies for analyzing and automating business processes emerged that are significantly changing the way how business process management is performed. The automation of business processes, which was mainly performed by workflow systems, is being revolutionized by RPA [LW]. In the field of BPI, data-driven techniques like process mining provided new ways in how to analyze and optimize processes [Ca09]. In the following section, we shortly introduce both techniques and motivate the development of DAM as an approach that connects data-driven process analysis and process automation. In addition, we discuss selected research that is in the broadest sense related to our approach.

2.1 Business Process Automation

A classical approach for process automation is the use of workflow applications to partially or completely automate a process [VdA13]. Realizing process automation involves integrating the workflow application with other systems such as databases and ERP systems. The integration process involves implementing new interfaces which the workflow application can use to communicate with other systems. Therefore integration efforts of workflow applications represent a major barrier in successfully realizing process automation [Ho97]. Furthermore, these workflow applications have to be configured, whereas the question of how to automate business workflow configurations has been a long-standing research effort [Li10].In recent years, RPA, which is a novel approach to automate business processes, is experiencing a rapid industry-wide adoption. Reasons for this are success stories such as [LWC15] which reports an impressive 3-year return-on-investment of 650-800 percent, whereas the investment represents the introduction of RPA. The advantage of RPA, compared to classical workflow systems, is that it can be easily integrated into a company's existing software systems landscape. While the utilization of workflow applications often involves an extensive integration effort, RPA systems can be integrated in a highly unobtrusive way [LW] since RPA interacts with software on the presentation layer. Today's state of the art RPA software mainly relies on two different ways to configure a software robot, whereas the first approach is predominantly used [LW]. The first approach is based on the manual configuration of software robots by describing the process flows and its rules. A user can define the process step by step, often using a tool which resembles a visual programming language. This configuration step results in an informal business process model, which doubles as process documentation. However, the manual configuration is time-consuming and requires detailed knowledge of the business process at hand. The second approach is based on software which records a single user's actions and attempts to replay them. However, since the software robot is based on only a single recording it is not robust against environmental changes e.g. varying directory content arrangements or changing application structure.

2.2 Process Mining

Process Mining is a research field which focuses on utilizing event logs in order to discover, analyze and optimize processes [VDA12]. While classical data mining techniques are often only used to analyze a specific step in a process, PM aims at analyzing the whole end-to-end process [VDA12]. PM is made possible by information systems in an organization which create and store the necessary event logs. Event logs are a collection of timestamped activities, recorded by a software system and usually executed by a user. Generally, an activity has multiple attributes, such as a resource attribute that specifies by whom the process was executed. Major challenges when applying PM are usually related to the event logs. Examples are incomplete and inaccurate data or heterogeneous data sources requiring extensive data-integration efforts. Challenges related to PM are further discussed in section 4.2.

2.3 Web Mining & Clickstream analysis

Classical PM is not the only field which utilizes data created by applications during their run-time. Web mining, specifically web usage mining, utilizes web transaction data and focuses on the discovery of user patterns in respect to website and web application usage [CMS97]. A common data source to perform various analyses are clickstream logs. A clickstream log consists of a sequence of web-page requests, which are usually recorded by a web-server. An application of clickstreams is the discovery of distinct usage patterns of web pages [LDM05]. However, since most business processes often involve more than one application, applying process mining on logs of only one application will not result in the complete picture of the business process. But using logs of various applications involved in a single business process introduces further challenges, such as discovering correlations between events of different data sources [Mo11]. [Ja10] discusses utilizing web usage mining to predict the future movement of users. A further application of click-stream analysis is to analyze user behavior on a web page with the goal to improve the design of the website [TKK05]. Moreover, [Wa13] present approaches to detect Sybil accounts by analyzing click-streams. These examples show that user click-streams are rich in information. However, since web-page requests represent the basis for the data, the presented analyses are restricted to web-based applications. Furthermore, click-stream analysis is not yet performed in the context of BPM.

3 Desktop Activity Mining

In the following section, we present the new concept of DAM which we position as a subfield of BPM that complements and interconnects PM and BPA. We define Desktop Activity Mining as "a method to record user activities on the detail level of desktop actions and to reconstruct and consolidate the resulting process variations with process and data mining techniques to discover an integrated process model."

The general approach of DAM is to record the desktop activities of all users executing a certain business process. Desktop activities, in this case, refers to user actions such as mouse-clicks, text input via keyboard, i.e. all actions that a user can perform when using a computer. The user activities data is combined with data from additional sources in the executed process, e.g. transaction data from information systems. The data of each process instance is consolidated and transformed in a format that can be used for further analysis. Process discovery algorithms are then applied to mine a consolidated process model from the various process instances. The resulting process model describes, on the one hand, the corresponding business process in an abstract way, and on the other hand, includes information down to the detailed level of user desktop activities. One of the main application scenarios for DAM is the configuration of RPA systems. In this case, the consolidated process model, including the detailed information about the user activities, needs to be transformed so that it can be used as a configuration input for the software robot. This is possible because all information that a robot needs to execute a certain action is available and the underlying process with its variations and exception is described. Thus, the software robots can be set up to perform the process in the same way the human employees would execute it. In this scenario, DAM offers a data-driven solution to configure RPA systems by analyzing the real process executions performed by human employees.

Due to its' process discovery component, DAM is closely connected to the concept of PM and, at least partially, uses similar algorithms to discover process models. However, in contrast to classical PM applications that typically rely on data from transaction logs of information systems, DAM builds upon the actual user activities, independently whether they are performed in information systems or not.

3.1 Reference Architecture

A DAM framework consists of two main components: First, a recording component that gathers data in different detail levels during the execution of each process instance. Second, an analysis component consolidating the information about different process instances and discovering the underlying process model. The recording component itself has to combine several functions in order to obtain an integrated picture of the full business process:

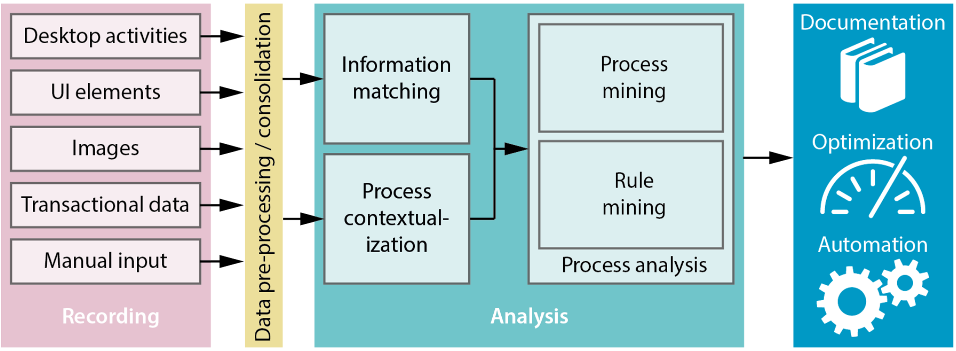

Fig. 1: Schematic overview of a Desktop Activity Mining Architecture

3.1.1 Desktop Activities Recording

Recording the basic user activities builds the ground of the DAM approach and is the first module of the recording component. In a typical office environment, users have two interfaces to interact with the computer: mouse and keyboard. Consequently, the user interaction with these devices needs to be recorded in all variations to gain a full picture of the process. In practice, this encompasses the click type and absolute location of a mouse-click and the numerical and textual input with the keyboard in addition to control keys, shortcut key combinations etc. Furthermore, it is necessary to determine the software-application that is active while executing a mouse or keyboard action. All events are recorded with a timestamp so that activities can be sequentially aligned.

3.1.2 UI Element Identification

The desktop activities, especially the mouse-click actions, are recorded with respect to a specific UI component. The position of mouse clicks per se do not contain information about the action that was triggered with the click. For later automation purposes, however, this information is essential. The UI Element Identification module extracts information from the active applications in which a desktop activity was performed. User actions in web browsers, for example, could be identified by and related to the addressed DOM element. In practice, however, most of the software does not provide information about its internal elements. The UI Element Identification module is therefore only applicable to a limited (although widely used) set of applications. In order to collect similar information from other SW applications, the only solution is to use image recognition techniques.

3.1.3 Image Recording

The image recording module extracts image patterns from the user's desktop. To realize this functionality, each user action is accompanied by a screenshot that documents the graphical user interface at the time when the action is performed. Depending on the further usage, different types of screenshots can be relevant. For documentation purposes, the full screen of the user can be recorded during execution time. To identify the graphical elements that are triggered by the user action, a small excerpt around the position of the user action can be sufficient. The identification of the graphical elements from the images is conducted later in an analysis stage.

3.1.4 Transactional Data Integration

Transactional data, obtained from information systems, can be used as a valuable addition to set the basic user activity data in the context of the underlying business process. As an example, the activity "Create Invoice" could be an event recorded from the invoicing software of the company when the user opens a new invoice template. All desktop activities after this event, describing the fill-out of the template are in the context of the same process step.

3.1.5 Manuel User Input

In order to mine processes by utilizing user-application interactions it is necessary to log every user-event, e.g. mouse-clicks and keyboard-inputs. However, it is difficult to derive meaning from these logs alone, since information about e.g. the currently active application and entered text does not unambiguously identify a step in a business process. Furthermore, the transactional data from the information systems can only provide limited information about the process context. Therefore, a module is required that allows users to manually define the starting or end-point of an underlying process step. This approach has the advantage that expert knowledge about the process is brought in by the user. The disadvantage is that it requires an additional manual effort during process recording and it is not robust against human errors and biases in interpreting the process.

The information that is recorded by the different recording modules is analyzed and consolidated in the subsequently discussed components.

3.1.6 Data Preprocessing / Consolidation

The key functionality of the recording component is to collect the necessary data from each process instance and each user that executes it. The output of the recording is, therefore, a collection of data sets that represent different process instances. The data consolidation and preprocessing component covers all the typical aspects of data preparation, i.e. noise reduction, outlier identification and removal, etc. Additionally, it transforms the data of different process instances into a logical structure and connects the information, e.g. by assigning IDs. Furthermore, it interconnects the different data types, e.g. screenshots, user action, etc., within a process instance and assigns multiple instances to a common business process.

3.1.7 Information Matching

The key concept of DAM is to capture the actions of multiple users and to integrate them in a consolidated process model. The recording component, however, captures the action of every single user with his or her individual desktop and software environment, e.g. a mouse click on a certain position of the desktop. The desktop arrangement of the different users is not necessarily identical, which means that a mouse-click of one user at a certain position does not necessarily reflect the same action as a mouse click of another user at the same position on his desktop. In case the active application provides information about the UI element that is triggered by the user action, identical activities of different users can easily be matched. In case the active application does not provide this information, the connection has to be derived by analyzing the recorded image patterns. The idea behind this is that for users performing identical actions, at least a part of the screens has to be similar, e.g. a group of buttons that appear in a software program. Image recognition can then be used to identify these similar patterns on the screenshots of each user. The localization of the user activities with respect to these patterns allows to relate them to a common process step.

3.1.8 Process Contextualization

The recording component records process information with different levels of detail. The lowest level being the user desktop activities, the middle level the transactions performed in information systems and the highest level the process knowledge provided by the users. The different information streams enable a dynamical and hierarchical view of the business process. In the Process Contextualization module, the information from the different streams is connected to obtain an integrated business process. The connection between the different data components is performed via the timestamp attribute that relates the events in a sequential order and enables a clustering of connected events. In a subsequent analysis component, this set of information enables a dynamical inspection of the process depending on the actual needs.

3.1.9 Process Analysis

The process analysis module combines the methods and algorithms that are used for analyzing and interpreting the recorded data. PM techniques are used to discover process models based on the data. Due to the different types of recorded data, different process models can be created that represent different detail levels of the process. In fact, the connection between user activities, transactional data and manually defined process steps enables the creation of an integrated, hierarchical process model that supports the navigation and inspection of the mined process in different layers. An additional important part of the process analysis is the determination of rules that define the execution paths. A rule, in this case, defines a correlation between different process steps and / or the data that is related to them. Knowledge about the correlations is especially important in case DAM is used as input to configure automation tools.

Finally, the DAM approach can deliver valuable input for different applications in the BPM context. First, it provides a way to obtain a detailed and automated process documentation. Second, the information can be used as input for process and resource optimizations. Finally, the structured UI activities and the derived process model and its rules can be used as input for the configuration of automation systems, thus reducing the required manual effort. We provide a discussion about possible applications in Sec. 4.

3.2 Desktop Activity Mining vs. Process Mining

Desktop Activity Mining and Process Mining are two different approaches for data-driven process discovery and documentation. On the one hand, they are closely connected, e.g. when process mining algorithms are used to analyze the recorded desktop activities. On the other hand, there are significant differences between Desktop Activity Mining and the general process mining approach.
The event logs to perform process mining are usually, as previously mentioned, obtained from information systems such as ERP and CRM systems. However, end-to-end processes often comprise more than one information system [BG11]. Thus, as long as these logs are not combined, process discovery results in a process model which only covers the interaction with a single system. Yet, combining event logs from different data sources can be a challenge, since different systems might log events in different granularity levels or capture different data attributes. Furthermore, not all systems log their events or the provided log data may be inaccurate or incomplete [Ge07], which introduces further challenges. DAM, on the other hand, provides event data that is complete and consistent throughout the whole process execution. Thus, DAM is expanding the possibilities of process mining

by providing a novel data source which offers a complete, homogeneous and uniformly high-quality event log, whereas PM is providing valuable process discovery algorithms to DAM. We performed a thorough comparison of both concepts in regard to *Data Source*, *Data Structure*, *Scope of Data*, *Goal*, *Requirements* and *Detail Levels*. The results are summarized in Table 1.

	Desktop Activity Mining	Process Mining
Data Source	The user-application interaction during a user's process execution path. Additionally, the user as source of knowledge. Furthermore, information systems which log transactional data.	Information systems which log transactional data.
Data Structure	Uniform over one DAM implementation.	Potentially heterogeneous data structures through heterogeneous information systems.
Scope of Data	Complete list of user-application interactions and user defined steps that are part of the process.	Transactional data which is recorded by Information Systems
Goal	Automation, analysis, optimization, documentation	Analysis, optimization, conformance checking
Requirements	DAM Software. Optionally information systems which provide transaction logs.	Information systems which create and store fairly high quality event logs.
Detail Levels	Two dimensional data structure consisting of user defined process steps and user-application interaction. Additionally a third, optional layer, the transaction log layer.	One-dimensional level of activities.

Tab. 1: Structured comparison of Desktop Activity Mining and Process Mining.

4 Proof-of-Concept and Discussion

To demonstrate the soundness of our conceptual artifacts we realized a software prototype which implements a selection of the modules defined in Section 3:

Desktop Activities Recording: The Desktop Activities Recording module records the user-application interactions. Here we can differentiate between mouse and keyboard events. Every time a mouse or keyboard event is triggered, a new action is recorded.

Manual Input: As previously discussed, manual user input is required to enrich the recorded user-application interactions with the business process context. Therefore, we provide the user with the functionality to type in a new process step or to select an already created process step.

Data Preprocessing / Consolidation: As mentioned in Section 3, elements of a process instance record such as screenshots, user actions and user-defined process steps have to be set in relation with each other. Most of this association is performed during process execution time since most of the required information is available during this period.

Process Analysis: We collected data by recording multiple instances of the process which we further discuss in Section 4. In order to analyze the mentioned data, four steps have to be performed. The first step is to extract the data related to a process from the central database and to store it in an appropriate format. The second step consists of selecting the hierarchy level which is of interest. In the third step, the data column that represents the states of interest has to be chosen. The fourth step consists of a process mining application to mine and analyze a corresponding process model.

To evaluate the prototype we tested it with an artificial scenario that reflects a typical process in customer relationship management. The process is initiated by the creation of a customer account. The new customer data needs to be entered in two software systems: a legacy accounting software that is only accessible via a terminal interface and a customized browser-based CRM system. After the registration of the customer, the information is sent to a sales accountant to trigger an initial customer meeting. The execution of the process requires the use of common software like Microsoft Outlook, old legacy systems with a minimalistic user interface and customized business information software. In addition, several different tasks from mouse-clicks to text inputs, key shortcuts etc. were possible.

After testing the prototype, we collected feedback from two user groups. First, the group of end-users who execute the processes. Second, the group of process managers who profit from the results of DAM. The feedback can be summarized in the following statements:

End-User point-of-view:

- The usage of the manual user input function is straightforward and easy to integrate into the execution of a process.
- The manual input is a manageable effort for users for a few process executions, but needs to be optimized for real and permanent operations.
- The acceptance and usability of the manual input function depends on the design of the user interface.

Process Manager point-of-view:

- The automated recording and consolidation of the process has the potential to reduce the effort for process documentation and automation.
- The connection between the information of different process detail levels is a valuable

feature, especially for process managers that need to drill-up or drill-down depending on the actual problem.

- The manual input function gives users freedom in respect to the granularity of process steps. This is not necessarily wanted for later process analysis. Providing usage instructions to the users might be an option.
- The manual input function requires the users to have a certain level of understanding of the process. This can not always be assumed.

In the future, Desktop Activity Mining can be relevant for several application scenarios. For process documentation, Desktop Activity Mining offers the possibility to document existing processes in an automated way and in much higher detail than currently possible. The process mining component allows a documentation of the process in an abstract process model, on the level of user desktop activities but also in a higher business context. In addition, it is documented which activities are required to execute the process. Together with the recorded screenshots / UI images, it enables persons to reproduce all possible process variations without knowledge about the process itself. Especially for the introduction of new employees, the process documentation based on DAM can be a significant support and reduce the period of adjustment. For process analysis applications, Desktop Activity Mining builds the ground to generate detailed insights into the business process. The different information layers enable a precise analysis of the process, e.g. its bottlenecks and faults, in a new level of detail. The effect of optimization efforts can not only be analyzed on a business process KPI level, but also through the actual changes in the employee's work. Finally, the implementation barrier for automation projects can be significantly reduced, since DAM collects and consolidates all information that is required to configure an automation task. In particular, for RPA SW, which partially already supports a configuration via recording a single-users process execution, DAM provides the information to automatically configure a software robot.

5 Conclusion and Future Work

The emergence of new technologies in BPM results in a need for a detailed discovery and description of office processes. In particular, RPA as a method to easily automate office processes requires detailed process knowledge to configure the software robots. A data-driven way to capture the required data from real process execution would significantly reduce the time and effort for calibrating and configuring such automation systems. Existing approaches for data-driven process discovery, such as PM, are not suitable to fulfill the requirements of these systems. Typical limitations are the level of detail of the analyzed process data or the restriction to a limited set of software applications. In this paper, we introduced Desktop Activity Mining as a new concept for BPM with the aim to record and analyze user activities in office processes which provides a new level of detail. We performed a detailed and structured comparison to the classical approach of PM and highlighted differences and advantages. With the presentation of a reference architecture and a prototypical implementation, we realized a proof-of-concept that shows the principal feasibility. A detailed evaluation with simulated and real office processes shows the potential of the approach to improve process documentation, optimization and automation in an office environment. As a next step in the further development of the approach, we plan to connect DAM with an RPA system and thus close the bridge between process discovery/documentation and the configuration of the automation software. Furthermore, the sequence of not all activities in desktop processes is predetermined and well structured, making process discovery more difficult and potentially leading to overly complex process models. Thus relevant PM research, such as [SGV09], should be incorporated. Future work will also have to include the investigation of Artificial Intelligence methods in order to develop software robots that are self-learning based on the data delivered by DAM.

References

[BG11] Bhat, J. M; Goel, S.: Mining Transaction Data for Process Instance Monitoring in Legacy Systems. In: AMCIS. 2011.

[Ca09] Castellanos, M; De Medeiros, AK Alves; Mendling, J; Weber, B; Weijters, AJMM: Business process intelligence. In: Handbook of research on business process modeling, pp. 456–480. IGI Global, 2009.

[CMS97] Cooley, R.; Mobasher, B.; Srivastava, J.: Web mining: Information and pattern discovery on the world wide web. In: Proceedings of Ninth IEEE International Conference on Tools with Artificial Intelligence. IEEE, pp. 558–567, 1997.

[Ge07] Genrich, M.; Kokkonen, A.; Moormann, J.; Zur Muehlen, M.; Tregear, R.; Mendling, J.; Weber, B.: Challenges for business process intelligence: Discussions at the BPI workshop 2007. In: International Conference on Business Process Management. Springer, pp. 5–10, 2007.

[HNP08] Hägele, M.; Nilsson, K.; Pires, J. N.: Industrial robotics. In: Springer handbook of robotics, pp. 963–986. Springer, 2008.

[Ho97] Hollingsworth, D.: Workflow-A Model for Integration. ICL SYSTEMS JOURNAL, 12:213–232, 1997.

[Ja10] Jalali, M.; Mustapha, N.; Sulaiman, Md N.; Mamat, A.: WebPUM: A Web-based recommendation system to predict user future movements. Expert Systems with Applications, 37(9):6201–6212, 2010.

[JN14] Jeston, J.; Nelis, J.: Business process management. Routledge, 2014.

[LDM05] Lu, L.; Dunham, M.; Meng, Y.: Discovery of significant usage patterns from clusters of clickstream data. In: Proc. of WebKDD. Citeseer, pp. 21–24, 2005.

[Li10] Liu, G.; Liu, X.; Qin, H.; Su, J.; Yan, Z.; Zhang, L.: Automated realization of business workflow specification. In: Service-Oriented Computing 2009 Workshops. Springer, pp. 96–108, 2010.

[LW] Lacity, M.; Willcocks, L.: , What Knowledge Workers Stand to Gain from Automation. https://hbr.org/2015/06/what-knowledge-workers-stand-to-gain-from-automation. Accessed: 2018-03-16.

[LWC15] Lacity, M.; Willcocks, L. P; Craig, A.: Robotic process automation at Telefonica O2. 2015.

[Mo11] Motahari-Nezhad, H. R.; Saint-Paul, R.; Casati, F.; Benatallah, B.: Event correlation for process discovery from web service interaction logs. The VLDB Journal—The International Journal on Very Large Data Bases, 20(3):417–444, 2011.

[SGV09] Song, M.; Günther, Ch.; Van Der Aalst, WMP: Trace clustering in process mining. Lecture Notes in Business Information Processing, 17 LNBIP:109–120, 2009.

[TKK05] Ting, I-H.; Kimble, Ch.; Kudenko, D.: UBB mining: finding unexpected browsing behaviour in clickstream data to improve a Web site's design. In: Proceedings. The 2005 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE, pp. 179–185, 2005.

[VDA03] Van Der Aalst, WMP: Business process management demystified: A tutorial on models, systems and standards for workflow management. In: Advanced Course on Petri Nets. Springer, pp. 1–65, 2003.

[VDA12] Van Der Aalst, W.: Process mining: Overview and opportunities. ACM Transactions on Management Information Systems (TMIS), 3(2):7, 2012.

[VdA13] Van der Aalst, WMP: Business process management: a comprehensive survey. ISRN Software Engineering, 2013, 2013.

[Vo04] Von A., R Hevner; March, S. T; Park, J.; Ram, S.: Design science in information systems research. MIS quarterly, 28(1):75–105, 2004.

[Wa13] Wang, G.; Konolige, T.; Wilson, Ch.; Wang, X.; Zheng, H.; Zhao, B.Y.: You Are How You Click: Clickstream Analysis for Sybil Detection. In: USENIX Security Symposium. volume 9, pp. 1–008, 2013.

[WLC15] Willcocks, L. P.; Lacity, M.; Craig, A.: Robotic process automation at Xchanging. 2015.