
Production Systems and Information Engineering
Volume 11 (1), pp. 29–47.
https://doi.org/10.32968/psaie.2023.1.3

ROLE OF USER ACTIVITY MONITORING IN RPA PROCESSES

PÉTER MILEFF
University of Miskolc
Hungary Institute of Information Technology
mileff@iit.uni-miskolc.hu

Abstract. User activity monitoring, process mining, and robotic process automation (RPA) are three related fields that have gained increasing attention in recent years due to their potential applications in various domains, including business process management and automation. User activity monitoring involves the use of software tools to track and analyze the activities of users on computer systems and networks, process mining aims to extract insights and knowledge from event logs of business processes, and RPA involves the use of software robots to automate repetitive and rule-based tasks. By combining these three approaches, researchers can gain valuable insights into how users interact with computer systems and networks, how business processes can be optimized and automated, and how software robots can be used to automate tasks and processes. In this paper, we focus mainly on the user activity monitoring, where several important factors and related problems are presented. The purpose of this publication is to review the field of user activity monitoring from several perspectives. We cover possible implementation alternatives, and the area of activity logs is presented in detail. The basic elements are presented that are necessary to create logos at a level that can later be suitable for analyzing and optimizing processes.

Keywords: User activity monitoring, RPA, Process mining

1. Introduction

In today's modern world, companies use complex IT systems to support their business processes. It is not uncommon to have a business where many people perform daily activities. A common characteristic of these systems is that dozens of different types of processes are performed by employees. Over time, the development of the company results in an increase in the complexity of the processes. Transparency and comprehensibility decrease, and often imperceptible bottlenecks can develop over time. The management of such complex systems raises

more and more problems both in terms of safety technology and in terms of process traceability. User Activity Monitoring (UAM) and RPA solutions, which have been developed as a result of recent times, try to find solutions to these problems.

User activity monitoring solutions are the use of software tools that monitor and track end-user behavior on devices, networks, and other company-owned IT resources.

It is advisable to classify UAM systems into two groups according to the application goals:

User monitoring in order to preserve information, security, and prevent malicious abuse

Many organizations use user activity monitoring tools to detect and prevent insider threats, whether unintentional or malicious. The range of monitoring and applied methods depends on the company's objectives. By implementing activity tracking, organizations can more effectively monitor employee online activities, ensure proper use of resources, and stop any suspicious activity that leads to a decrease in productivity and endangers the organization. They can more easily identify suspicious behavior and mitigate risks before they result in a data breach, or at least in time to minimize damage.

Monitoring of user activities to support RPA processes

In this case, the purpose of collecting information is to create an information data set that can be the basis for a later RPA or Process Mining analysis and practical implementation. Its main feature is the accurate and well-structured storage of information. The extracted data set provides an opportunity to explore bottlenecks using graph theory methods, to automatically reproduce processes, to support them with artificial intelligence, and to predict potential events.

The aim of this paper is to provide a detailed overview of the role of user activity monitoring, to show the standards that are available and to propose a direction that can provide suitable results during practical application as well.

2. Literature overview

In the context of process mining, data preprocessing techniques can be categorized differently based on various criteria. The existing event log preprocessing techniques can be categorized into two main groups: transformation techniques and detection-visualization techniques.

Figure 1 [17] provides a schematic representation of a possible taxonomy. This proposed taxonomy effectively organizes the diversity of existing preprocessing techniques and allows for the identification of common characteristics among them.

The first category encompasses techniques that apply transformations to the event log to correct imperfect behaviors, such as handling missing, irrelevant, or duplicate

data, before implementing a process mining algorithm. On the other hand, the second category consists of techniques designed to detect or diagnose imperfections in the event log. While the second category only identifies potential problems related to data quality, the techniques in the first category directly address and rectify the identified imperfections in the event log.

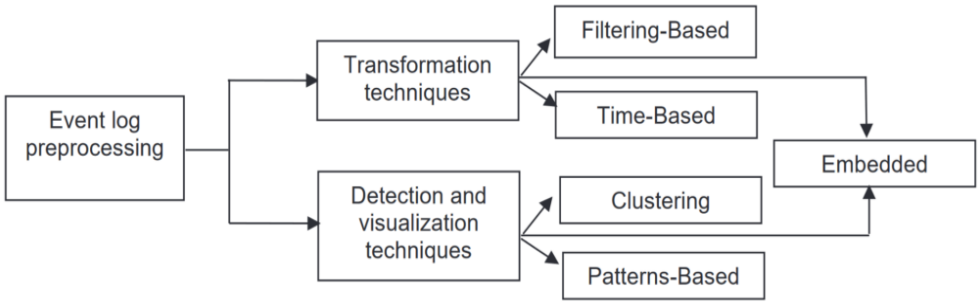


Figure 1. Event log preprocessing techniques [17]

Transformation techniques involve performing operations and actions on the raw event log's original structure to enhance its quality. Within this category, two main approaches are utilized: filtering and time-based techniques. Filtering techniques are focused on evaluating the likelihood of event or trace occurrences based on their surrounding behavior. Events or traces with low frequencies of occurrence are eliminated from the original event log, aiming to remove logging mistakes and prevent their propagation to the process models. These techniques primarily address noise, anomalies, and missing values in the event log.

There are various works in the literature which have proposed the development of filtering techniques. For example, Conforti et al. [18] introduced a technique that identifies anomalies in a log automaton. This technique first abstracts the process behavior recorded in the log as an automaton, capturing direct follow dependencies between events. Subsequently, infrequent transitions are removed using an alignment-based replay technique while minimizing the number of events removed from the log.

Another approach, proposed by van Zelst et al. [19], is an online/real-time event stream filter designed to detect and eliminate spurious events from event streams. This approach relies on dominant behavior, which attains higher occurrence probabilities within the automaton compared to spurious behavior. The filter was implemented as an open-source plugin for both ProM [20] and RapidProM [21] tools.

In their research, Wang et al. [22] focused on techniques to recover missing events, thereby providing a collection of more complete provenance candidates. The authors employed a backtracking approach to reduce redundant sequences associated with parallel events. To implement this, they introduced a branching framework, allowing each branch to apply backtracking directly.

On the other hand, Niek et al. [23] proposed four innovative techniques aimed at filtering out chaotic activities. Chaotic activities are activities that lack clear positions in the event sequence of the process model and have unchanging (or minimally changing) probabilities of occurrence as a result of the occurrences of other activities. In other words, these chaotic activities do not follow the usual process flow.

Cheng and Kumar [24] had the objective of building a classifier using a subset of the log and applying its rules to eliminate noisy traces from the log. They put forth two proposals. The first proposal involved generating noisy logs from reference process models and applying process mining algorithms to both the noisy log and its sanitized version. By comparing the discovered process models with the original reference model, they could assess the effectiveness of their approach. The second proposal entailed comparing the models obtained before and after log sanitization, utilizing structural and behavior metrics for evaluation.

In contrast, Mohammadreza et al. [25] proposed a filtering approach based on conditional probabilities between sequences of activities. Their method estimates the conditional probability of an activity's occurrence based on the number of preceding activities. If this probability falls below a certain threshold, the activity is considered an outlier. The authors classified both noise and infrequent behavior as outliers in their approach.

Detection-visualization technique is a different approach. Within this category, two main approaches are recognized: clustering and pattern-based techniques.

Clustering techniques segment the event log into multiple subsets, facilitating the examination and analysis of each subset separately. The subsequent step involves pinpointing noise or anomalous elements within the scrutinized subsets. Clustering is among the most frequently used data preprocessing techniques in process mining, primarily utilized for detecting quality issues related to noisy values and data diversity. Through the formation of similar clusters, it becomes possible to identify patterns of imperfections associated with noisy data in various attributes of the event logs.

Within the realm of detection-visualization techniques, some of these techniques involve preparing event logs by identifying patterns based on heuristic rules. These rules are derived from observed behaviors or the expertise of analysts in process mining who have studied various event logs from different domains. Many pattern-based techniques assert that the event log may not be entirely accurate if a specific pattern is not detected within it [26]. These techniques often work alongside clustering, abstraction, or alignment techniques to identify patterns related to noisy data or data diversity.

Suriadi et al. [26] propose a method to assess event log quality by describing a collection of eleven log imperfection patterns derived from their experiences in preparing event logs. These patterns are abstract representations of recurring behaviors in specific non-arbitrary contexts.

Ghionna et al. [27] describe an approach that combines the discovery of frequent execution patterns with a cluster-based anomaly detection procedure. The method

employs specialized algorithms to reduce the occurrence count of spurious activities and to cluster a log and its associated S-patterns (patterns and clustering) simultaneously.

In contrast, WoMine-i [28] extracts infrequent components from the logs using model specifications (e.g., task sequences, selections, parallels, loops, etc.). WoMine-i utilizes an a priori search that starts with minimal patterns and prunes the infrequent patterns to reduce the search space.
Jagadeesh et al. [29] propose an iterative method for transforming traces by identifying looping constructs and sub-processes, which are then replaced by abstracted entities in the event log.

2.1. Tools are available for the event logs preprocessing

According to Will van der Aalst [30], process mining tools that involve event log preprocessing can be classified into three categories:

- C1: process mining tools are primarily designed for addressing ad-hoc questions related to event log preprocessing. For instance, Disco [89] is an example of such a tool, allowing users to interactively filter the data and promptly project it onto a newly learned process model.
- C2: process mining tools, the analytical workflow is explicitly defined, empowering users to visualize and make decisions regarding the isolation or elimination of specific elements from the event log. RapidProM is an example of a tool falling under this category.
- C3: tools are tailored to repeatedly answer predefined questions in a known setting. These tools are commonly used to create process dashboard(s) that offer standard views of process models. Celonis Process Mining is an example of a tool that supports the creation of such process-centric dashboards.

Event log preprocessing tools are typically integrated into certain process mining software suites. However, many commercial process mining software tools, like Perceptive Process Mining by Lexmark, Interstage Business Process Manager Analytics by Fujitsu Ltd., Minit by Gradient ECM, myInvenio by Cognitive Technology, and others, do not offer built-in support for event log preprocessing tasks aimed at enhancing the quality of event logs.

To address this gap, specific tools, applications, or frameworks have been developed to cater to distinct event log preprocessing tasks. These tools often focus on specific process modeling languages and rely on various data deployment or transformation techniques. On the other hand, specialized tools like ProM [20], Apromore [31], Celonis [32], and RapidProm [33] have been designed to include diverse filters, routines, and algorithms that facilitate preprocessing of event logs to support process mining tasks. These tools offer a more comprehensive approach to handling various preprocessing requirements effectively.

Table 1. Comparison of popular process mining tools [17]

Features	Prom 6.5.1	Apromore	RapidProM	Disco	Celonis
Trace/event Filtering	Yes	Yes	Yes	Yes	Yes
Trace clustering	Yes	Yes	Yes	No	Yes
Timestamp repair	Yes	Yes	Yes	Yes	No
Remove attributes, events or activities	Yes	Yes	Yes	Yes	No
Embedded preprocessing	Yes	No	Yes	No	No
Abstraction	Yes	No	No	No	No
Alignment	Yes	Yes	Yes	No	No

3. Operation of user activity monitoring

We usually use some kind of activity monitoring software to monitor user activities. The activity monitoring software records the use of applications and programs on the monitored workstation. User activities displayed on the screen are recorded in a pre-developed and well-structured log. Logs are thus information databases that store all the activities that happened that day. Thanks to today's modern technology, there are many options and solutions available for monitoring, monitoring and managing activities. Some important techniques:

- **Logging and analysis:** the most common form of user activity monitoring. In this case, we store the events in a classic text file or database. All events and processes are well defined. The resulting log file stores the information in a well-structured form for later processing and analysis. Perhaps we can say that it serves as the basis for several other applied techniques.
- **Video recordings of the sessions:** user activity is recorded in the form of a video stream. An important goal is the possibility of subsequent analysis, the prevention of possible security abuses, and the possible reproduction of the interactions carried out.
- **Capturing a screenshot:** similar considerations make up the goal of this technique as in the video-based approach. With this method, we record images, which provide a basis for further analysis. In several practical implementations, the UAM tool on the captured screenshots is images for framing the given activity with a box. For example, the item selected by the user, buttons pressed, etc.

- **Logging of keyboard usage:** logging of pressed keys can be important in cases where the corporate or other system used by users expects less graphical interaction, more processes may require manual data entry/typing.
- **Network packet inspection:** perhaps more of a solution belonging to the security group of UAM systems. Logging of packets sent and received by users and users of the system. Traffic control offers many data protection options for the company, it can even protect against potential abuse.
- **Kernel monitoring:** low-level monitoring of the kernel of the operating system. Its use is also mainly justified in security matters.

All collected information should be examined within company policies and user role boundaries to determine related processes, activities to be analyzed later, and potential unwanted user activities. What "inappropriate user activity" actually means always depends on the company. It is up to the company to select UAM monitoring solutions that can cover anything from visiting personal websites or shopping during business hours to stealing sensitive company data such as intellectual property or financial information.

3.1. User activity monitoring tools

Many tools can be used to monitor or support user activity. These tools range from general security software applications to dedicated tools for monitoring sessions and activities, creating a complete logging trail for each user. Also known as privileged account security solutions, there are tools designed to monitor and secure privileged account activities and centralize policy management.

The best user activity monitoring tools include real-time monitoring systems. These tools log user activities in the background in real-time (virtually instantaneously), monitor the user's screen in the background, and when targeted activity occurs, alert the administrator or manager or take appropriate actions. These can be simple, for example, suggestions to support the operator, or perhaps the execution of automated background processes at some level. In other cases, even reporting suspicious activities to IT and security teams. Without real-time, risks may go unnoticed in some cases while the responsible IT group deals with other known issues. Thanks to today's technology, it is not necessary for an entire IT team to monitor and analyze activities. Developing a good UAM solution that supports monitoring, analysis, reporting and possible intervention of user activity greatly simplifies the problem.

4. Operation of user activity monitoring

The most important starting point for any activity that later wants to build on the results of user activity will be the so-called event log. The technique most often used in practice is Process Mining, which aims to analyze data from a process perspective. It looks for answers to questions such as "What is the current state of the process?",

"Are there unnecessary steps that could be eliminated?", "Where are the bottlenecks?" and "Are there any deviations from the established and prescribed process rules?".

In order to be able to analyze the data in the event log in any form, they must be written to the log using some process-specific approach. This will make it possible to later decide on the step in the event log to which process it belongs and which step of the given process it was.

Perhaps one of the most important questions of the topic may be, "Where does the data come from in the event log?"

The answer to the question immediately divides the information source and the activity monitoring method into two groups.

4.1. Systems providing logs and event histories

Systems that already include event logging in themselves, which are usually very configurable in many cases. For example, customer relationship management (CRM), IT service management (ITSM), order management, and workflow systems typically fall into this category. From the point of view of activity monitoring and process mining, these systems belong to the easier side of the implementation spectrum, since the data is most likely already available in the form of logs, data tables (history tables) and other solutions, and they can often be used directly. For example, in an ITSM system there is a ticket number in every case, and all status changes are recorded.

In many cases, this data can be easily exported as a CSV file and directly imported into the pipeline mining tool without any prior processing.

4.2. Systems that require human activity to build history

Here, at the more difficult end of the spectrum, we find database-centric systems such as Enterprise Resource Planning (ERP) and certain custom or legacy systems. In these systems, event histories are not available in a ready-to-use form, because they were not planned and designed that way from the beginning. Therefore, in each case, it is necessary to prepare some unique solution for finding the relevant data in the business database tables and creating our own event log for analysis.

An additional challenge for these systems is if the process is supported and used by several IT systems. A typical example would be when purchase requests are managed in the ERP system, and invoices are supported through a separate financial system. If we want to merge data from two (or more) systems, the most important thing to pay attention to is how we can track the case on these different systems. For example, if the ERP system always uses a purchase order number and the financial system always uses an invoice number, then a reference to the order must be found in the invoice.

The solution for these systems can be divided into two ways:

- **Completing the source code of the system:** one of the best solutions can be if we have the source code of the system and place event logging in the code for required and critical points from the point of view of process analysis. Since the entire source code can be modified, the event logging can be completely solved. The rather big disadvantage of this method is that it is often dangerous to repeatedly modify the source code of a well-functioning system and the complete retesting of the system can be time-consuming. This depends on how the previous development environment was designed, whether some automatic testing procedure was developed, etc. Development can be further complicated by the fact that in many cases, although the source code is available, many years have passed since the system was developed, so updating the source code and individual supporting function libraries is essential. Finally, the modified system has to be put into operation (again).
- **Application of user activity monitoring software:** another alternative is to use existing activity monitoring software. The advantage of the solution is that it is not necessary to modify the code or any part of the operating system, but only the UAM software needs to be installed and configured on the systems and stations to be monitored. Today's modern UAM software can more or less manage/log many types of user interaction. Unfortunately, we cannot say that the problem has been fully resolved, for the following reasons: although user interactions are logged, the binding of events to process steps cannot be fully realized in most cases. The reason for this is that the event that occurred – be it a pressed button – must be provided with a unique identifier that can be linked to the process as one of its steps. Since the identifiers of the UI elements were determined during development, they may not have been designed to be suitable for later logging.

In practice, due to the above, engineers mostly try to use some kind of UAM software to implement the solution because the source code is not available, or the software was created by another company. It is important to mention that determining which UAM software we use depends on the implementation of the software we want to log, which can be thick or thin client-based. While in previous years, IT software was typically developed using thick client methods, today we mostly encounter thin client, web-based solutions. The two approaches require a different UAM solution. For thick clients, there is hardly any UAM software on the market. In this case, the software is highly dependent on the operating system, making it extremely difficult for an external software to log the desired other software. In many cases, saving screenshots is the solution, but this can also raise many additional problems (resolution, lack of UI identifiers, etc.).

Fortunately, the world's shift towards thin clients has opened up new opportunities in the field of logging as well. Many software packages and plug-ins are available to achieve the desired result.

5. Problems and obstacles related to implementation

A well-implemented RPA system usually requires a complex implementation process. Many sub-tasks, which can be complicated in themselves, by their very nature hide uncertainties and problems. In the following, we highlight some of the more important problems of this nature.

5.1. Lack of activity logs

As mentioned earlier, the right data set is one of the key issues for IT systems based on artificial intelligence, as it enables the design of learning algorithms and the proper training.

Perhaps one of the most important questions is whether there will be a set of data of sufficient quality during the project? The question expresses a legitimate concern, since a system that was put into operation years ago was probably not designed in such a way that user activities can be easily extracted from it. In addition, the various components of a multi-component system do not behave in the same way, they have different structures, which certainly differ in their logging activity. It is a natural requirement that during the project, the creation of activity logs with appropriate detail should be included in the partner's systems, however, since this is a time-consuming task in itself, it is not expected that usable activity logs will be available from the start.

The sample system must therefore be designed in such a way that it is able to generate data that is close to or equal to reality. In this way, the lack of the initial data set can be solved, research tasks and the design of algorithms are not blocked.

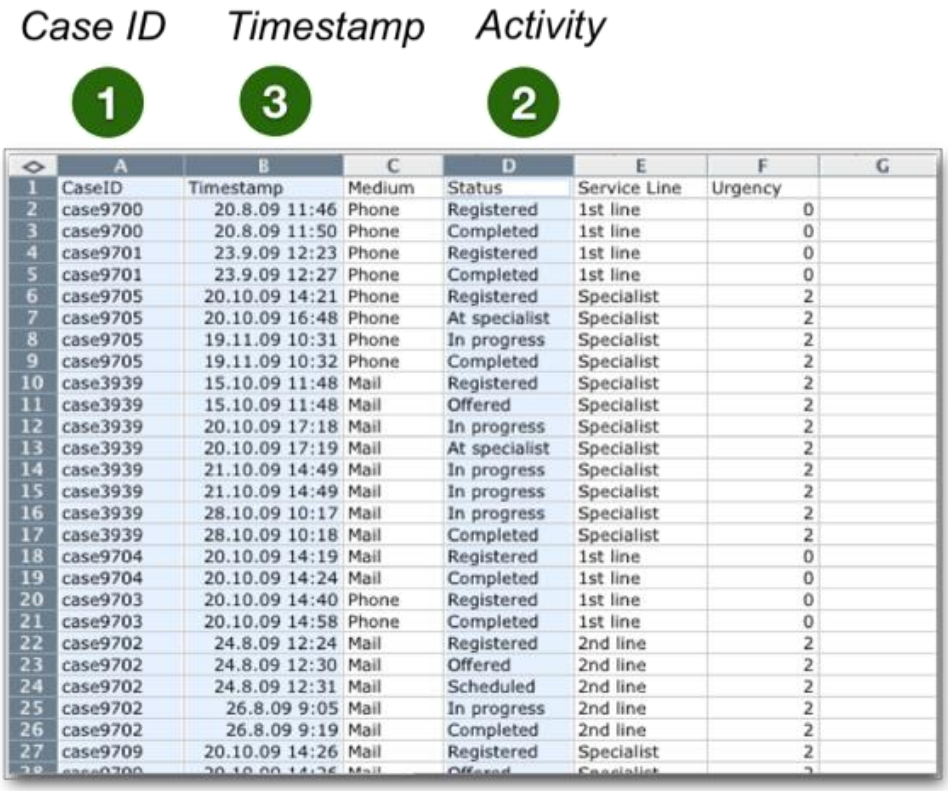
5.2. Shortcomings of activity logs

An activity log must basically consist of three mandatory elements: case id, activity and a time stamp. Each case represents a step in an execution instance of a process. For example, in a product ordering process, handling the order represents one case. It is a very important criterion to know which process belongs to which case.

Activity is one step in the process. For example, a document creation process may consist of the following steps: "Create", "Update", "Submit", "Approve", "Request Revision", "Review", "Publish", "Reject" (different people such as authors and editors). Some of these steps may happen more than once in a case, but not all of them need to happen every time. The different process steps or state changes performed during the process must be given appropriate names. If there is only one entry (one line) in each case, the data is not detailed enough.

The data must be at the transaction level (access to the history of each case) and must not be aggregated into a single case. Also, be aware that the chosen activity affects the level of detail with which we view our process. Sometimes it may be worth defining the name of the activity as a combination of several columns, and there may be several alternative views of what constitutes an activity. Finally, a third

important prerequisite for process mining is to have at least one timestamp column indicating when each activity occurred. This is important not only for analyzing the timing behavior of the process, but also for determining the sequence of activities in the event log. Sometimes each activity in the process has a start and an end timestamp. This is good because it allows you to analyze the processing time of an activity (the amount of time someone actively spent doing that task), which is known as execution time or also called activity management time.



	A	B	C	D	E	F	G
1	CaseID	Timestamp	Medium	Status	Service Line	Urgency	
2	case9700	20.8.09 11:46	Phone	Registered	1st line	0	
3	case9700	20.8.09 11:50	Phone	Completed	1st line	0	
4	case9701	23.9.09 12:23	Phone	Registered	1st line	0	
5	case9701	23.9.09 12:27	Phone	Completed	1st line	0	
6	case9705	20.10.09 14:21	Phone	Registered	Specialist	2	
7	case9705	20.10.09 16:48	Phone	At specialist	Specialist	2	
8	case9705	19.11.09 10:31	Phone	In progress	Specialist	2	
9	case9705	19.11.09 10:32	Phone	Completed	Specialist	2	
10	case3939	15.10.09 11:48	Mail	Registered	Specialist	2	
11	case3939	15.10.09 11:48	Mail	Offered	Specialist	2	
12	case3939	20.10.09 17:18	Mail	In progress	Specialist	2	
13	case3939	20.10.09 17:19	Mail	At specialist	Specialist	2	
14	case3939	21.10.09 14:49	Mail	In progress	Specialist	2	
15	case3939	21.10.09 14:49	Mail	In progress	Specialist	2	
16	case3939	28.10.09 10:17	Mail	In progress	Specialist	2	
17	case3939	28.10.09 10:18	Mail	Completed	Specialist	2	
18	case9704	20.10.09 14:19	Mail	Registered	1st line	0	
19	case9704	20.10.09 14:24	Mail	Completed	1st line	0	
20	case9703	20.10.09 14:40	Phone	Registered	1st line	0	
21	case9703	20.10.09 14:58	Phone	Completed	1st line	0	
22	case9702	24.8.09 12:24	Mail	Registered	2nd line	2	
23	case9702	24.8.09 12:30	Mail	Offered	2nd line	2	
24	case9702	24.8.09 12:31	Mail	Scheduled	2nd line	2	
25	case9702	26.8.09 9:05	Mail	In progress	2nd line	2	
26	case9702	26.8.09 9:19	Mail	Completed	2nd line	2	
27	case9709	20.10.09 14:26	Mail	Registered	Specialist	2	

Figure 2. A sample log file that contains the three required elements

5.2.1. Issues related to case ID

Case ID must be combined from multiple fields: This happens when the main case ID alone does not uniquely identify the process instance. For example, if we analyze the tax filing process at the IRS, each citizen is identified by their social security number. However, tax returns are prepared every year. If we have data from multiple years, we must link the year for which the statement was submitted in addition to the social security number. When creating the activity log, we should therefore not forget all the fields necessary to identify the individual case.

Use different case identifiers in different parts of the process: If multiple identifiers are used throughout the process, we need to be able to follow the chain of a case from start to finish. For example, if in a purchasing process incoming requests are identified by a PO number, and later payment activities are linked to an invoice number, there is a need for the invoice identifier to be linked to the order number (PO), to have some reference to help in coordinating processes. When extracting activity, a persistent case identifier is needed in the log to serve as a reference for each step in the process.

Multiple case IDs have many-to-many relationships: If different case IDs are used in a process, they may not always have a 1-to-1 mapping (for example, two orders combined in one delivery, or one order in two pieces). In this case, it is necessary to determine from which point of view we wish to examine the process. It may be necessary to create multiple logs/data exports if multiple viewpoints are to be analyzed.

5.2.2. Activity-related problems

The second minimum requirement for logs is the mentioned activity. Activities are different steps or state changes in a process. IT systems can record not only activities that are important to us, but also less interesting debugging information. That is why it is important to make sure that the logging also records the relevant activities. The existence of less relevant activities in the log is not a problem in itself, they can be filtered out later.

It is very important to name the activities correctly from the beginning. If there are several candidates for the name of the activity, it is advisable to keep them all, because during the analysis it is possible to examine different aspects. For example, during a bank loan application process, the system can record changes in both internal status (showing the stage of the internal process) and external status (showing the status of the customer or sales channel). Both aspects can be useful. If we keep both fields in the data, we can enjoy full flexibility in the analysis later.

It is important that the names of the activities recorded in the log are easily readable by humans. Most processes are complex, and we won't get very far in the analysis if the steps in the process map are just technical status numbers or operation codes. Of course, we can also record technical operation codes in the data, which can later be useful for retrieving technical questions related to the data. However, the human-readable name is not negligible, because it specifies exactly what the status means. These human-readable state names should be included as separate columns in the data.

Activities can be hidden in several places. In database-centric systems, certain activities can often be extracted from various business data tables. In this case, all the events of the process are not automatically visible. In such cases, it is necessary to define the most important milestone activities and then collect the activities based on them.

5.2.3. Time stamp issues

The third minimum requirement is the time stamp. Each activity requires at least one timestamp to sequence each event. And if we want to analyze the duration of the activity, then a start and an end time stamp must be entered for each activity.

Efforts should be made to record the timestamps as accurately as possible. For example, if the hours and minutes data are available, it is not enough to just record the date, but it is also advisable to store the time. And if seconds or milliseconds are available, it's even more successful even if it doesn't seem like that much detail is needed for analysis. Creating detailed timestamps can be useful for properly sequencing activities.

More than two timestamps per activity: In some cases, not only one or two timestamps are available per activity, but more. For example, you might have a timestamp that indicates whether a task is ready to be recorded, a second that indicates when the person started working on it, and a third that indicates that the task was completed. It is advisable to record them all. It may be useful later on for various questions to be answered. For example, what is the average time between done and completed.

The date and time are listed in separate columns: Since we can record several time stamps at the same time, it is advisable to include the date and time of a single time stamp as one field. This will help in later analysis.

Different timestamp patterns: Timestamps from different data sources may have different formats. For example, one activity may have a datetime stamp format of 01SEP2013, while another activity may have a datetime stamp format of 2013-09-10 07.36.51.711899. A uniform timestamp format is expected in the diary. If several formats are available, we either standardize them or treat them as separate data.

Timestamp Override: Certain timestamps may be overwritten during repetitive activities. When activity timestamps are stored in a database, it is often assumed that the process is executed in the ideal order and no reprocessing occurs. Of course, this is incorrect in most situations: Rework may occur, for example, because this task was not performed the first time and the corresponding process step must be repeated later. If the system does not record all timestamps of activity repetitions (often keeping only the most recent timestamp), then valuable information is lost.

5.3. Well-known errors in logs

Any deficiencies in the input data, the so-called "holes", fundamentally affect the final result. The most common problems are the following:

- **Formatting errors:** One of the most common errors is that, in some cases, the format of a given line of the log is different from the prescribed one. This can

greatly affect the result of the analysis, in the best case it will only result in a processing error. Special characters, an unnecessary quotation mark can cause many headaches.

- **Missing events:** even if the data has been imported without errors, there may still be problems with the data. One typical problem is the lack of data and events. There are two types:
  1. **Holes in the timeline:** sometimes unusual gaps are discovered in the number of events occurring within certain time frames of the log
  2. **Unexpected amount of data:** it is necessary to roughly know the number of activities each process consists of. Due to an unexpectedly large amount of data, some of it may be lost in certain parts of the processor without any kind of indication due to a simple design error.

Missing activities: Some activities of the process may not be recorded in the data. There may be manual activities (such as phone calls) that people do at our desks. These activities occur during the process but are not visible in the data.

Missing Timestamp: In some situations, we know whether an activity has taken place or not, but there is no timestamp. Maybe the activity came from another system. In such cases, the data processing module cannot determine the location of the event in the timeline.

Unfinished processes: the log also contains processes that have a "start" signal, but no "end" signal. The cause of the problem can be found in the following:

1. The data extraction method only examines events within a certain time interval. For example, some processes started in the previous year (before January) and continued in the following year. In this situation, you will only see the part of these processes that occurred in the year you are analyzing.
2. Some processes have not been completed yet. Even if all data has been extracted, some cases may not be completed yet. They are still "somewhere in the middle".
3. Some processes never finish. Not all processes will have an "end" signal. In some cases, even due to a technical error, the process is interrupted. For example, after making a payment in a webshop, the customer is not directed back to the store's interface. These cases do not end at any of the expected endpoints, they never end.

5.4. Solution for the problems

As we can see, many problems can arise during the processing and operation of activity logs. Every problem requires a unique solution, it may depend on the processes to be executed, so there is no general solution key. Developing an efficient

event log and process mining system requires a lot of attention and careful planning on multiple levels in order to make it "foolproof". There are three levels of planning:

- **Filesystem level:** it includes the determination of the exact format of the event logs and their possible extensions. Unification of logs from different systems, clear storage scheme.
- **Validation level:** Validation, filtering and possible correction of inputs. It is inevitable that errors do not occur during the operation of a system, no matter how well-designed the system is. Robust operation, preparation for exceptional events.
- **AI based level:** The highest level of processing should be provided by a processing system supported by artificial intelligence. These systems are capable of correcting a specific error, inferring a missing data, thus even working in the form of incomplete data and remaining functional.

  By including the three levels and possibly additional aids, event logs can be processed efficiently, and errors during operation can be handled well.

6. Solving the problem with a neural network

Artificial neural networks are one of the most researched areas today. With its help, problems become solvable that were not possible before, or only partially so. Of course, the development of hardware provided a great clue to this. Today, there are many models in the literature, the most effective approaches are often related to the so-called deep learning field. In many cases, the training of artificial neural networks is reduced to a supervised regression problem, but classification and unsupervised training are also feasible. Two phases can be distinguished in the operation of the networks: in the training phase, given the known input parameters and expected outputs, the weights are changed in such a way as to minimize the value of a loss function (for example, the mean square error). In the prediction phase, the trained neural network then forms an output upon passing an unknown input, which can be, for example, the probability of belonging to a category.

In practice, neural networks are often used to predict different activities. Based on the data of the activity log, the sequence of operations can be identified, and the current sequence of activities can be matched to the existing patterns. However, the neural network-based approach to process discovery raises a number of problems. In no way can it be called a typical ANN problem, because solving the problem requires unsupervised learning, and neural networks are mainly suitable for supervised learning. In this area, many research directions/problems will come to the fore, which will seek the answer to the question of how to create a network that is able to solve the mentioned problem. In the future, the development of a new neural network model with a dynamic structure will form the basis of the research.

6.1. Teaching pattern

The learning process of neural networks requires training samples. The lack/ inadequate quality of the previously mentioned input data can fundamentally affect the

learning process. The input data set must be sufficiently complex – including all occurring processes and their activities – so that the various approaches, hypotheses and models can be fully tested during the model creation. It can be further stated that a single sample data set is also not sufficient by itself, otherwise the developed model cannot be tested effectively in dealing with the diversity that occurs in reality.

6.2. Resource requirement

In recent years, the so-called LSTM-type neural networks, which have proven their effectiveness in countless places (Siri, Cortana, Google voice assistant, Alexa), are playing an increasingly prominent role. Although they are effective, it is advisable to pay attention to their disadvantages:

The learning process of LSTM-based networks is more difficult because they require a calculation properly adjusted to memory bandwidth, which is a separate headache for hardware designers and limits the practical applicability of the network. LSTM needs 4 linear layers (MLP layers) per cell to run at each sequence time step and each stage. Linear layers require a large amount of memory bandwidth, as a result, they often cannot use many computing units, because the system does not have enough memory bandwidth to "feed" the computing units. Adding additional computing units is easy, but increasing memory bandwidth is much more difficult. As a result, RNN / LSTM and variants cannot be properly accelerated in hardware.

So, working with LSTM-based networks in any case results in a greater need for resources, which mainly affects the learning phase. In addition, the set of data handled and expected as input in connection with the project is also large. It is therefore necessary to develop an environment for researchers where they can use the available server capacities in a shared manner. The sample system has sufficient memory, computing and storage capacity. In this way, instructors do not use their own (usually low-performance) machine to test more complex teaching processes, but an environment that provides results much faster.

7. Conclusion

The administrative processes of modern companies are becoming more and more complex, which can be supported with increasingly complex systems. Of course, as in all other areas, the question of automation arises here, with the help of which complicated processes can be partially simplified. RPA is just such an initiative, during which, by logging and monitoring user activity, an automated decision support system can be built that is capable of solving or even predicting certain steps in the complex process. Due to the fact that the area is quite modern, it has not yet matured, and there are no boxed products available for implementation. In this publication, we have summarized the most important requirements and typical problems of the field, which provide a good basis for the implementation of such a system.

References

[1] van der Aalst, W.: *Process Mining: Data Science in Action*. Second Edition. Springer Berlin, Heidelberg, 2016.

[2] van der Aalst, W. M. P., Bichler, M. & Heinzl, A.: Robotic Process Automation. *Bus. Inf. Syst. Eng.*, 60, pp. 269–272, 2018. https://doi.org/10.1007/s12599-018-0542-4

[3] Aguirre S, R.: Automation of a business process using robotic process automation (RPA): a case study. *Appl. Comput. Sci. Eng. Commun. Comput. Inf.*, 2017. https://doi.org/10.1007/978-3-319-66963-2_7

[4] Hofmann, P., Samp, C. & Urbach, N.: Robotic process automation. *Electron. Markets*, 30, pp. 99–106, 2020.

[5] Fung, H. P.: Criteria, use cases and effects of information technology process automation (ITPA). *Advances in Robotics & Automation*, 2013.

[6] Berruti, F., Nixon, G., Taglloni, G., & Whiteman, R.: *Intelligent process automation: The engine at the core of the next-generation operating model*. 2017.

[7] Leopold, H., van der Aa, H., & Reijers, H. A.: Identifying candidate tasks for robotic process automation in textual process descriptions. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W. & Bera, P. (eds.). *Enterprise, business-process and information systems modeling*. Lecture notes in business information processing. Vol. 318, pp. 67–81, Springer International Publishing, Cham, 2018.

[8] Moffitt, K. C., Rozario, A. M. & Vasarhelyi, M. A.: Robotic process automation for auditing. *Journal of Emerging Technologies in Accounting*, 15 (1), pp. 1–10, 2018. https://doi.org/10.2308/jeta-10589.

[9] Penttinen, E., Kasslin, H. & Asatiani, A.: How to choose between robotic process automation and Back-end system automation? In: *Proceedings of the 28th European Conference on Information Systems (ECIS)*, Portsmouth, UK, 2018.

[10] Vedder, R. & Guynes, C. S.: The challenge of botsourcing. *Review of Business Information Systems (RBIS)*, 20 (1), 2016. https://doi.org/10.19030/rbis.v20i1.9677

[11] Jorge, R., Rui, L., Tiago, E., Sara, P.: Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review. *Procedia Computer Science*, Vol. 181, pp. 51–58, 2021.

[12] Leno, V., Dumas, M., La Rosa, M., Maggi, F. M. & Polyvyanyy, A.: *Automated Discovery of Data Transformations for Robotic Process Automation*. 2020 https://arxiv.org/abs/2001.01007

[13] Huang, F., Vasarhelyi, M. A.: Applying robotic process automation (RPA) in auditing: A framework. *International Journal of Accounting Information Systems*, 35, 2019.

[14] Siderska, J.: Robotic Process Automation — a driver of digital transformation? *Engineering Management in Production and Services*, Vol. 12, No.2, pp. 21–31, 2020. https://doi.org/10.2478/emj-2020-0009

[15] Ivančić, L., Suša Vugec, D., Bosilj Vukšić, V.: Robotic Process Automation: Systematic Literature Review. In: *Business Process Management: Blockchain and Central and Eastern Europe Forum*. BPM 2019. Lecture Notes in Business Information Processing, Vol. 361, Springer, Cham, 2019.

[16] dos Santos Garcia, C., Meincheim, A., Faria Junior, E. R., Dallagassa, M. R., Vecino Sato, D. M., Carvalho, D. R., Santos, E. A. P., Scalabrin, E. E.: Process mining techniques and applications – A systematic mapping study. *Expert Systems with Applications*, Vol. 133, pp. 260–295, 2019.

[17] Marin-Castro, Heidy M., Tello-Leal, Edgar: Event Log Preprocessing for Process Mining: A Review. *Applied Sciences*, Vol. 11, Issue 22, p. 10556. 2021. https://doi.org/10.3390/app112210556

[18] Conforti, R., Rosa, M.L., ter Hofstede, A. H. M.: Filtering Out Infrequent Behavior from Business Process Event Logs. *IEEE Trans. Knowl. Data Eng.*, 29, pp. 300–314, 2017.

[19] van Zelst, S. J., Fani Sani, M., Ostovar, A., Conforti, R., La Rosa, M.: Filtering Spurious Events from Event Streams of Business Processes. In: Krogstie, J., Reijers, H. A. (eds.). *Advanced Information Systems Engineering*. Springer International Publishing, Cham, Switzerland, pp. 35–52, 2018.

[20] van Dongen, B. F., de Medeiros, A. K. A., Verbeek, H., Weijters, A., van der Aalst, W. M.: The ProM framework: A new era in process mining tool support. *International Conference on Application and Theory of Petri Nets*. Springer, Berlin/Heidelberg, Germany, pp. 444–454, 2005.

[21] van der Aalst, W. M. P., Bolt, A., van Zelst, S. J.: *RapidProM: Mine Your Processes and Not Just Your Data*. arXiv 2017, arXiv:1703.03740.

[22] Wang, J., Song, S., Zhu, X., Lin, X., Sun, J.: Efficient Recovery of Missing Events. *IEEE Trans. Knowl. Data Eng.*, 28, pp. 2943–2957.

[23] Tax, N., Sidorova, N., van der Aalst, W. M. P.: Discovering more precise process models from event logs by filtering out chaotic activities. *J. Intell. Inf. Syst.*, 52, pp. 107–139, 2019.

[24] Cheng, H., Kumar, A.: Process mining on noisy logs-Can log sanitization help to improve performance? *Decis. Support Syst.*, 79, pp. 138–149, 2015.

[25] Sani, M. F., van Zelst, S. J., van der Aalst, W. M. P.: Improving Process Discovery Results by Filtering Outliers Using Conditional Behavioural Probabilities. In: *Proceedings of the Business Process Management Workshops-BPM 2017 International Workshops*, Barcelona, Spain, 10–11 September 2017, pp. 216–229.

[26] Suriadi, S., Andrews, R., ter Hofstede, A. H., Wynn, M. T.: Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Inf. Syst.*, 64, pp. 132–150, 2017.

[27] Ghionna, L., Greco, G., Guzzo, A., Pontieri, L.: Outlier Detection Techniques for Process Mining Applications. In: *Foundations of Intelligent Systems*, Springer: Berlin/Heidelberg, Germany, pp. 150–159, 2008.

[28] Chapela-Campa, D., Mucientes, M., Lama, M.: Discovering Infrequent Behavioral Patterns in Process Models. In: Carmona, J., Engels, G., Kumar, A., (eds.). *Business Process Management*. Springer International Publishing, Cham, Switzerland, pp. 324–340, 2017.

[29] Jagadeesh Chandra Bose, R. P., van der Aalst, W. M. P.: Abstractions in Process Mining: A Taxonomy of Patterns. In: Dayal, U., Eder, J., Koehler, J., Reijers, H. A., (eds.). *Business Process Management*. Springer: Berlin/Heidelberg, Germany, pp. 159–175, 2009.

[30] van der Aalst, W. M. P.: *Process Mining-Data Science in Action*. 2nd ed.; Springer, Berlin/Heidelberg, Germany, 2016.

[31] T. A. Foundation: *Apromore: Advanced Process Analytics Platform*. The University of Melbourne, 2011. Available online: https://apromore.org/(accessed on 19 April 2021).

[32] Celonis, S. E., Munich, G.: *Celonis Process Mining*. CELONIS, New York, NY, USA, 2009. Available online: https://www.celonis.com/ (accessed on 19 April 2021).

[33] Mans, R., van der Aalst, W., Verbeek, H.: Supporting process mining workflows with RapidProM. In: Limonad, L., Weber, B. (eds.). *Proceedings of the BPM Demo Sessions 2014 co-located with BPM 2014*, Eindhoven, The Netherlands, 20 September 2014, pp. 56–60.