



Process Discovery Analysis for Generating RPA Flowcharts

Fabian Rybinski^(✉)  and Selina Schüler^(✉) 

Institute of Applied Informatics and Formal Description Methods (AIFB),
Karlsruhe Institute of Technology (KIT), Kaiserstr. 89, 76133 Karlsruhe, Germany
{fabian.rybinski,selina.schueler}@kit.edu

Abstract. Robotic Process Automation (RPA) is a frequently used approach for automation in IT systems. RPA uses existing graphical user interfaces to automate simple, rule-based tasks from a user's perspective. Before automation can be accomplished using RPA, extensive knowledge about individual user interactions must be gathered. This information is represented in a flowchart by a human designer, which is then typically refined in a trial-and-error approach. This approach becomes time-consuming and error-prone as processes become more complex.

In parallel, Process Discovery techniques as part of Process Mining enable the automated generation of process models from event logs. Thus, they are considered appropriate to simplify and automate the described creation of flowcharts in the context of RPA. In this regard, the so-called UI logs are particularly relevant. These can be used to record user interactions with various user interfaces within a process.

This paper examines existing Process Discovery methods for application in the context of RPA. The goal is to clarify what is needed for the automatic creation of RPA flowcharts with Process Discovery on the basis of UI logs.

The results show that existing Process Discovery methods generate process models that are not suitable for immediate translation into RPA flowcharts, as they are mainly control flow oriented. For the creation of RPA flowcharts that are suitable for automation, the integration of linked data is fundamental. Therefore, a prototypical implementation demonstrates a method that enables the automated translation of a process model into an RPA flowchart.

Keywords: RPA support · RPA flowchart generation · Process Discovery

1 Introduction

Robotic Process Automation (RPA) automates process steps by using software robots that imitate the execution of manual tasks. For this purpose, process steps that take place on graphical user interfaces are recorded and automated. RPA focuses on simple, repetitive tasks that are executed across different IT

systems. Current RPA tools offer the ability (a) to drag and drop process paths into flowcharts by using simple rules such as if-then formulations, or (b) to generate such flowcharts by directly recording user interactions within a process path. A flowchart allows the representation of simple, sequential processes in which only one activity is executed at a time [21]. RPA can perform high-volume tasks with high accuracy, consistent quality and high efficiency. Developing RPAbots requires thorough knowledge of the tasks to be automated, the IT systems involved, and their user interfaces. This knowledge is currently obtained through interviews and recordings and then implemented in trial-and-error approaches [10]. This is time-consuming and error-prone and requires extensive testing, which is complicated by direct integration with live systems [12].

Process Discovery is a Process Mining technique to automatically build process models from process execution data recorded by IT systems [2]. Process Discovery enables the visualization of process data in models to identify improvement potentials with respect to certain attributes such as time. Therefore, the combination of these two research areas is obvious, as Process Discovery could automatically create the relevant process paths of a process for RPA. Process Discovery can provide information regarding important questions for RPA, such as how the process steps are ordered or which process paths are the most frequent or efficient. By creating a complete model with all paths, it becomes possible to assess all variants of the process in regard to the desired output. Although there is existing work on combinations of these topics, there exist several shortcomings: There is no work on which Process Discovery algorithm could be used to generate RPA flowcharts, what the requirements are for these algorithms, and which of these algorithms produce the best results. Furthermore, the existing work on implementing process steps with RPA does not enable the analysis and improvement of processes. As a result, RPA is often seen as a short-term solution. To enable a long-term use of RPA, the process should be analyzed in advance and not only one process variant should be implemented with RPA. Therefore, the decision options of alternatives should be implemented in the RPA solutions, so that several variants of a process are represented. In addition, these variants to be implemented should be freely selectable.

This paper is structured as follows. The second section introduces RPA and related flowcharts, Process Discovery and an overview of related work. Then the limitations and shortcomings of existing Discovery Algorithms for the application in the context of RPA are described to motivate the idea of the prototype. Next, the prototype is presented that generates RPA flowcharts based on UI logs. Finally, an evaluation is given, based on a use case, showing the potential of the prototype, but also some of the existing problems and future research opportunities.

2 RPA, Process Mining and Related Work

In this chapter, selected basics and examples of the areas of Robotic Process Automation and Process Mining are presented first. Afterwards, related work is introduced.

2.1 Robotic Process Automation (RPA)

Robotic Process Automation (RPA) is a frequently used term in the context of business process automation [34]. According to the Robotic Process Automation Institute's definition, the term RPA covers intelligent software and the application of that software [24]. Thus, RPA is a completely software-based solution of process automation. RPA uses the applications, functionalities and security of a company's existing IT systems, so there is no need for integration with those systems [36]. For this reason, RPA is described in literature as a lightweight automation method and is considered an evolution of macros and scripts [14]. The evolution consists in implementing more complex logics and processes. In doing so, RPA uses the existing graphical user interfaces for working with IT systems and thus only sits on top of these existing systems [6]. This enables RPA to (partially) automate cross-system processes without implementing interfaces.

At the beginning of RPA implementation detailed knowledge about tasks, involved user interfaces and user actions is necessary [9]. Currently, this information is obtained through interviews, workshops, or observation of user interactions. Based on this information, the RPA flowcharts are developed by a human designer [10]. In a trial-and-error approach, a flowchart is created first with all the activities that the software robot is supposed to perform. These activities have to be specified in great detail (e.g. write in Excel cell A3) and possible decisions in case of multiple paths have to be defined in advance [18]. The execution of the actions is then monitored and the flowchart is adjusted in the case of any errors. An exemplary RPA flowchart is shown in Fig. 1. Besides the sequence of the activities, the respective data of the activities must be known in order to create the flowcharts [31]. For example, this is the ID of a specific Excel cell or a browser text field that the robot is supposed to select in the execution.

RPA is suitable for automating extensive, repetitive tasks that are considered both time-consuming and trivial to perform. To be particularly suitable, these tasks need to fulfill the following characteristics: structured data, rule-based process and deterministic results [26]. Additionally, there are other criteria such as frequent execution, infrequent exceptional cases, different systems involved and vulnerability to human error [22]. Tasks that meet these requirements are often found in the so-called back offices, i.e. finance, purchasing or HR departments [13]. For example, the robot is capable of moving the mouse or copying data from one file to another. The capacities freed up by automation with RPA can be used by humans for more unstructured and interesting tasks. This might result in benefits in employee satisfaction as well as scalability and 24-h serviceability [37].

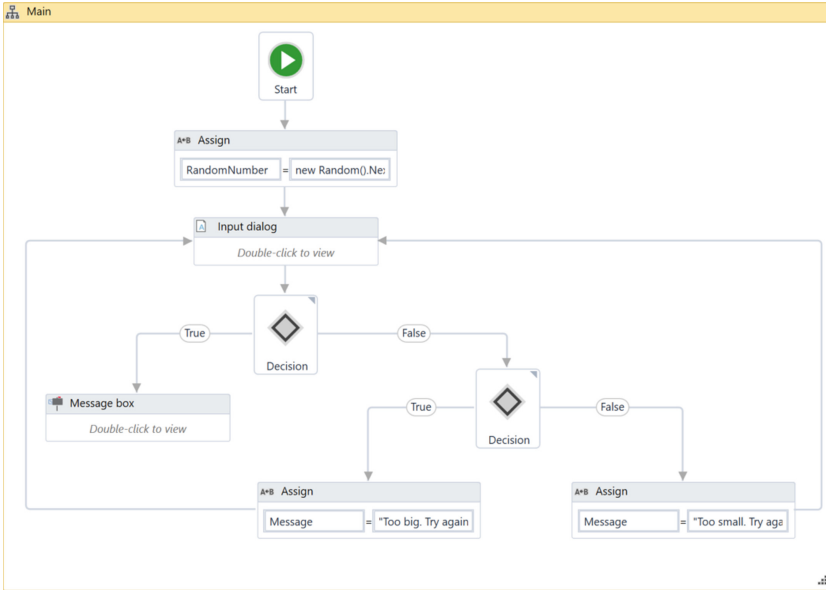


Fig. 1. RPA flowchart example [38]

2.2 Process Mining

Process Mining is a research area that links the areas of data science and process science [4]. The term Process Mining is used to describe techniques that allow the capture, analysis and improvement of real processes [5]. For example, Process Mining can be deployed to examine deviations from the to-be process or to analyze performance. Companies can use Process Mining to increase the understanding and control of the own processes. The gained insights are based on the real behavior reflected in the event logs and thus enable fact-based business process management [3].

Basically, the area of Process Mining is subdivided into three different categories: Process Discovery refers to the generation of process models, Process Conformance is the comparison of real process executions with modeled processes, and Process Enhancement adds new aspects to the existing model [2]. All three categories have in common that they build on event logs. These event logs record occurring events in IT systems and are often already stored in the company's IT systems (e.g. ERP system, CRM system) [8]. The information associated with an event, such as activity names, timestamps, and the person assigned to it, is recorded in the event logs [7]. For the application of Process Mining techniques they need to be extracted, pre-processed and combined. Fundamental to understanding Process Mining is the assumption that events are recorded sequentially in the event logs [1]. In addition, each event must correspond to a task and a case.

In the following sections, the focus is on Process Discovery as one category of Process Mining and, especially, on the existing algorithms in that category. The goal of Process Discovery is to generate a process model that represents the behavior in the event log without prior knowledge about this model [1]. Thus, the Process Discovery techniques provide the basis for further analysis of the processes within a company. In this context, a Process Discovery algorithm takes an event log with activities as input and generates a process model over the same activities [39].

2.3 Related Work

To gain insight in current approaches in the literature that address the research area covered, a *Systematic Literature Review* (SLR) was conducted. The objective was to identify literature that discusses Process Mining and especially Process Discovery in the context of RPA. This also involves identifying the requirements of process models in the context of RPA. Based on this objective, a search string was created containing terms from Process Mining, Process Modeling and RPA: (“RPA” or “Robotic Process Automation”) and (“Process Mining” or “Process Discovery” or “Process Learning” or “Workflow Discovery” or “Workflow Learning” or “Model” or “Flowchart” or “Script” or “UI Log”). A forward and backward reference search was then performed with the relevant publications identified. The literature research was conducted in December 2020. Based on the method described, a total of 14 publications could be classified as relevant for further analysis. The following section briefly summarizes current approaches to Process Discovery in the context of RPA.

Nowadays, Process Mining methods are already used in the early stages of the RPA lifecycle to support the human designer [25]. The selection of the process to be automated also takes place in these phases [15, 29]. In particular, Process Discovery could be appropriate to create process models for the automation with RPA. In general, event logs are used in the context of Process Discovery methods in order to represent the occurred process activities in a control flow model. In this context, the corresponding data (e.g. specific excel cell) that allows the execution of activities with RPA is especially important and therefore must be represented in a process model [31]. In [31] the authors describe the concept of a multi-perspective Process Discovery method that integrates the data flow and the control flow in a single process model. Furthermore, instead of the well-known event logs, the UI logs are used in the context of RPA, which contain granular user interactions, e.g. clickstreams, keystroke protocols [32]. Several approaches of UI loggers already exist in the literature and allow logging user interactions in e.g. Microsoft Excel or Internet browsers [33, 35]. The mentioned loggers produce logs that are applicable for further analysis with existing PM techniques.

In [11, 30] the authors describe more holistic approaches. The introduced software tools include UI loggers and allow the automatic translation of recorded sequences into RPA flowcharts with all necessary data assignments. In [11] the translated sequence is selected based on the shortest duration in the log and Process Discovery is only part of the graphical representation of the high-level

process. In [30] the tool allows the determination of automatable actions in the UI log and the subsequent translation in an RPA flowchart. Finally, in [23] the authors propose the automation of RPA itself, describing a self-learning approach that requires minimal human intervention.

Another related area is that of automated planning [17]. These techniques allow learning of operators that can model different instantiations of process models through automated planning, as an alternative to PM. With these models, very large-scale compositions can be generated and used to capture an RPA flowchart [19]. In the presented approach, each non-determinism accounts for a different instantiation.

Overall, the SLR showed that although theoretical research exists on combining RPA and PM, there is a lack of realization. Before automating a process with RPA, the analysis of the process is inevitable, which is why application of PM techniques seems to be promising. Therefore, this paper reviews whether it is possible to generate RPA flowcharts from UI logs based on Process Mining and which Process Discovery methods are most appropriate.

3 Existing Process Discovery Methods for RPA

In this chapter, existing approaches of Process Discovery methods and the corresponding algorithms are assessed and compared regarding their suitability for use in the context of RPA. It will be verified whether the existing algorithms are suitable to create process models based on UI logs. These process models should subsequently be able to be used for translation into an RPA flowchart.

In the literature, there is a wide variety of Process Discovery algorithms that can be used for discovering process models from event logs. Meanwhile, there are more than 40 different algorithms that tackle the discovery problem using different techniques. As the algorithms differ, so do the results, i.e. in the modeling language and thus in the resulting process models. For example, there are algorithms for generating Petri nets, BPMN models, directly-follows graphs, process trees, or Declare models. Some of the well known algorithms might be the *alpha-algorithm* [8], *Heuristics Miner* [40], *Inductive Miner* [27], or the *Evolutionary Tree Miner* [16]. The following section briefly summarizes the characteristics of the algorithms of the mentioned Process Discovery methods, including some evolutions.

The algorithm in [42] is an evolution of the original α -algorithm, which is one of the first Process Discovery algorithms in the literature [8]. The *alpha++* ($\alpha++$) algorithm allows the determination of non-free-choice constructs in process models.

In [41] the authors describe an evolution of the original heuristics miner [40]. The goal of the *Flexible Heuristics Miner* (FHM) is to fully utilize the advantages of the basic idea of the original algorithm. For this purpose, causal nets for the representation of process models are defined for the first time.

The *Inductive Miner - Infrequent* (IMi) [28] is an algorithm that generates process trees and allows the processing of infrequent behavior in the event logs.

The IMi ensures that the model is represented correctly and with high fitness, i.e. the process model should represent the behavior in the event log.

The *Evolutionary Tree Miner* (ETM) is based on a genetic approach [16]. The algorithm generates an initial population of process trees that represent the behavior in the log. In further steps, the initial population is continuously adapted until defined criteria in the four quality dimensions fitness, precision, generalization and complexity are met.

A closer look at the aforementioned Process Discovery algorithms reveals that they all discover only the control flow in the generated process models. In this context, control flow describes the order of the different activities of an event log [3]. The same applies to the other existing algorithms in the literature. Based on the approach of the SLR, no algorithms were identified that reflect the data and data transformation in the model or considered these data when generating the models. Since the existing algorithms only consider the control flow in the process models based on activity ID, they do not distinguish between same activities with different data assignments. Furthermore, as the data is not included when generating the models, it is not possible to assign the data to the activities afterwards. The models generated in this way are therefore not suitable for translation into an RPA flowchart that already contains the data required for automation. Thus, to enable the automated translation of a process model into RPA flowcharts, adjustments must be made.

4 Prototypical Approach and Architecture

As described in the last chapter, the current Process Discovery algorithms do not consider data assignments when generating process models. Thus, without modifications they are not suitable for generating process models that can be used to create RPA flowcharts. Since the assigned data is essential for execution within RPA, the prototype is intended to implement a solution that enables the generation of process models with the necessary data assignments. Ultimately, the creation of RPA flowcharts is desired that already contain the required data for automation. Generally various solutions are conceivable for achieving this goal. On the one hand, existing Process Discovery algorithms could be adapted or new ones developed. The development of algorithms that integrate the data in the generation of process models seems promising, e.g. multi-perspective process mining [31]. These enable the mapping of the data flow in addition to the control flow. On the other hand, the adaptation of the UI logs used as input for Process Discovery could also be goal-oriented. Existing algorithms distinguish activities mainly based on activity ID. Thus, to distinguish same activities with different data, the data could be noted in the activity ID. As an example, an activity *copy* could be extended by *cell A2* and could be clearly distinguished from the activity *copy cell A3*. Hereby, existing algorithms should distinguish the different activities based on the extended activity ID and map them to different nodes of the process model.

Comparing the options, the development of new Process Discovery methods is a sustainable solution. Here, the requirements for process models in the context of RPA can already be considered in the design of the method and the corresponding algorithm. Due to the scope of the paper at hand, this is not possible and is a future goal. However, the considerations for the adaption of the UI logs already seem to enable the required process models for a prototypical implementation. Additionally, the application of existing analysis tools to the discovered process models is possible and results of conducted evaluations of existing algorithms may be considered for the selection of the algorithm. Therefore, in the following sections, a prototype is designed and implemented to automatically generate RPA flowcharts based on Process Discovery.

4.1 Design and Implementation of the Prototype

For the presented prototype an adaption of the UI logs is chosen. That means that the UI logs should be adapted in such a way to enable the generation of process models with data assignments. To generate the process model, Process Discovery is applied on the adapted UI logs. Finally, based on the discovered process model, a single path of the model can be chosen and translated into an RPA flowchart. Consequently, the prototype is composed of three different parts, preceded by an existing UI logger and followed by an existing RPA tool. Its overall design is presented in Fig. 2.

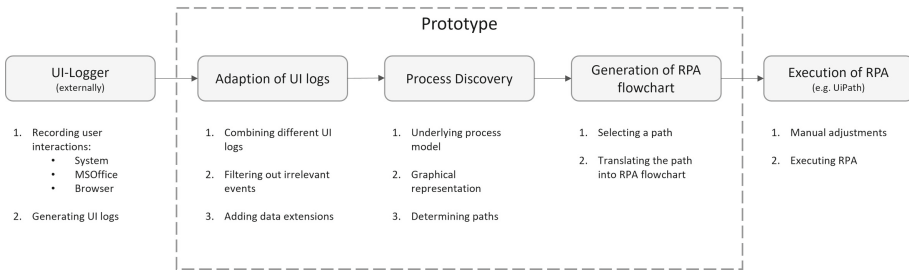


Fig. 2. Prototype structure

For recording user interactions the UI logger presented and implemented in *Smart-RPA* [11] is used. It allows the detailed recording of all relevant user interactions and especially the data necessary for the generation of an RPA flowchart, e.g. Excel cell, browser URL. The same paper is also the basis for the translation of events into RPA activities, but here adjustments have been made to build a prototype that enables the manual selection of a variant for automation based on the analysis of a given process model. This process model should include all data of a given UI log, as opposed to simply selecting the most frequent or shortest variant.

The prototype handles user interactions in the operating system, Microsoft Excel and Microsoft Edge. These already allow a variety of different actions and build the basis for the integration of further applications. In a first step, the UI logs have to be adjusted. The input of the algorithm consists of CSV files generated with the described UI logger. Each CSV file captures one instance of the process, since different instances cannot be distinguished in a single file. These are imported into a data frame and the column labels are matched to the XES standard, e.g. *time:timestamp*. Additionally, a *case:concept:name* column is created to represent the case ID. This is repeated for each CSV file and finally a combined data frame is created with all user interactions. The combined file captures all recorded instances of a process and is thus comparable to an event log. To enable the generation of process models with data assignments using existing Process Discovery algorithms, the column *concept:name* is extended with the relevant data from the other columns. Since Process Discovery algorithms generate process models based on this column, the data is included in the process model. Thus, it is possible to link it to a specific activity in the UI log. Since the goal is to distinguish between activities with the same activity name, the data in which the activities differ is added. For example, Excel activities can be distinguished by worksheet and cell, or browser activities by URL and element ID. The final data frame contains passed to the second component, which contains the Process Discovery. The compliance with the XES standard enables the application of traditional Process Discovery algorithms. According to this description, the algorithm presented in Fig. 3 results. Since the implementation of the prototype is done in Python, the implementation of the Process Discovery is done with PM4PY [20], which is a Process Mining library that provides Process Mining algorithms. From this library, the α -miner is chosen for Process Discovery in the prototype. The algorithm generates process models that are mainly low in complexity and therefore particularly suitable for graphical representation.

Algorithm 1: Adaption of UI logs

Procedure LogProcessing(*csv_list*):

for CSV-Logs in *csv_list*:

df \leftarrow Rename columns to match XES standard

df \leftarrow Create column *case:concept:name*

df \leftarrow Generate dataframe with user interactions of csv

end

combined_df \leftarrow Combine all user interactions in one dataframe

combined_df \leftarrow Create column *frequency*

for rows in *combined_df*:

if *activity* in *activity_list*:

combined_df \leftarrow Extend *concept:name* with data of relevant columns

Fig. 3. Adaption of UI logs

Even though incorrect models may arise with it, the α -miner is chosen as it provides a simple realization for the first implementation that should provide appropriate results. The generated process model serves as input for the generation of an RPA flowchart. Additionally, it is input for the graphical representation, which is a support for the selection of an execution path. To identify all paths in the process model, an algorithm is used that creates paths in a loop until each transition in the process model has been involved at least once. These paths are the basis for selection by the user. In case of only two different paths, they can both be transferred in an RPA flowchart together with an if-else relation.

The identified paths are passed to the third component of the prototype. Here, the frequency of the paths is determined by the minimum frequency of the involved nodes. This information, in addition to the graphical representation of the process, is another support for the user. After selecting a path, the activities of the selected path are translated into software code. The XAML format (Extensible Application Markup Language) is used, which can be read by common RPA tools. The translation of the activities is possible by adapting the UI logs, as this allows the representation of data in the process models and the distinction between activities. Thus, the data is correctly assigned to the activities in the flowchart and it can be executed in an RPA tool.

5 Evaluation of the Prototype

In this section, the prototype is tested. For this purpose, an exemplary use case and the resulting outcome of the prototype are presented first. Afterwards, the strengths and weaknesses of the prototype are discussed.

5.1 An Use Case in the Prototype

To evaluate the prototype, a test case is used that contains activities from the applications Microsoft Excel, Microsoft Edge and the operating system. In addition, several paths are included to test the identification of them. The test case contains three different paths and thus 2 different decisions in the process model. For this, the data of 3 different persons are to be transferred from an Excel file into a web form. A total of 15 different executions of the activities are recorded with the UI logger, the frequencies of the paths are 10, 4 and 1. The first component of the prototype correctly adapts the 15 UI logs and combines them in a CSV file. In this file, the XES standard is applied and the frequency column is created. Furthermore, the contents of the column *concept:name* are correctly extended with the respective data of the other columns, e.g. the activity *copy-Excel* with the cell, file and data sheet. A section of the combined CSV file is given in Fig. 4. Since the combined UI log contains the required adaptations, the next step is to start Process Discovery. A section of the generated process model is given in Fig. 5. The figure shows that 3 different paths and 2 decisions have been included in the process model. The intended information is visualized in the transitions. The label is composed of the node ID, the activity name and

case:concept:name	time:timestamp	org:resource	category	application	concept:name
413171032005000	2021-04-13T17:10:32.005	fabia	MicrosoftOffice	Microsoft Excel	openWorkbook Daten_Formular.xlsx
413171032005000	2021-04-13T17:10:39.504	fabia	Browser	Edge	open Browser
413171032005000	2021-04-13T17:11:02.410	fabia	Browser	Edge	autoBookmark https://www.google.de/
413171032005000	2021-04-13T17:11:09.375	fabia	Browser	Edge	autoBookmark https://www.umfrageonline.com/s/e6e569b
413171032005000	2021-04-13T17:11:14.130	fabia	MicrosoftOffice	Microsoft Excel	getCell A3 Daten_Formular.xlsx Personendaten
413171032005000	2021-04-13T17:11:16.571	fabia	Browser	Edge	clickTextField https://www.umfrageonline.com/s/e6e569b element-11-23828958
413171032005000	2021-04-13T17:11:16.717	fabia	MicrosoftOffice	Microsoft Excel	pasteExcel A3
413171032005000	2021-04-13T17:11:18.046	fabia	Browser	Edge	changeField_pasteExcel
413171032005000	2021-04-13T17:11:26.270	fabia	Browser	Edge	https://www.umfrageonline.com/s/e6e569b element-11-23828958
413171032005000	2021-04-13T17:11:28.420	fabia	MicrosoftOffice	Microsoft Excel	close Browser
					closeWorkbook Daten_Formular.xlsx

Fig. 4. Combined CSV file generated by the prototype

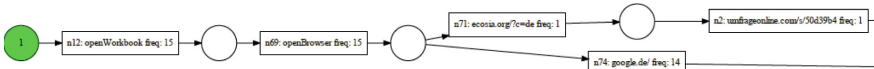


Fig. 5. Section of a process model in the prototype

the frequency. Especially the data relevant for RPA are represented and allow the translation into an RPA flowchart. In the given case, the most frequent one is selected for the generation of the RPA flowchart. With the third component of the prototype, the activities are translated into an XAML file. This file is then imported with the RPA tool *UiPath*. The tool translates the XAML file and represents it in a flowchart. A section of the generated flowchart is given in Fig. 6. The figure shows that the required data is correctly inserted into the respective activities. Finally, the flowchart is executed and the software robot transfers the personal data to the web form as intended.

5.2 Strengths and Improvement Potentials

The use case of the prototype has shown that the desired functions are achieved. In particular, it replaces the current trial-and-error approach in the application of RPA. The benefit results in the automated generation of a process model in which the data necessary for RPA is included. This process model simplifies the analysis of the processes to be automated. In addition, the current approach is improved by the automated generation of RPA flowcharts based on a process model. There is no need to manually transfer user interactions from UI logs to the flowchart and the analyst can focus on monitoring the execution and making any adjustments. In general, the prototype at this stage works particularly well for rather simple processes that contain clearly definable paths without loops.

Although the prototype implements the desired functions, some improvements are still necessary. Particularly, the complexity of the processes is currently a limiting factor. In the test, a process model with loops was created unintentionally. Due to the current algorithmic implementation, this can not be processed in a reasonable way at this stage of the prototype. Another limiting

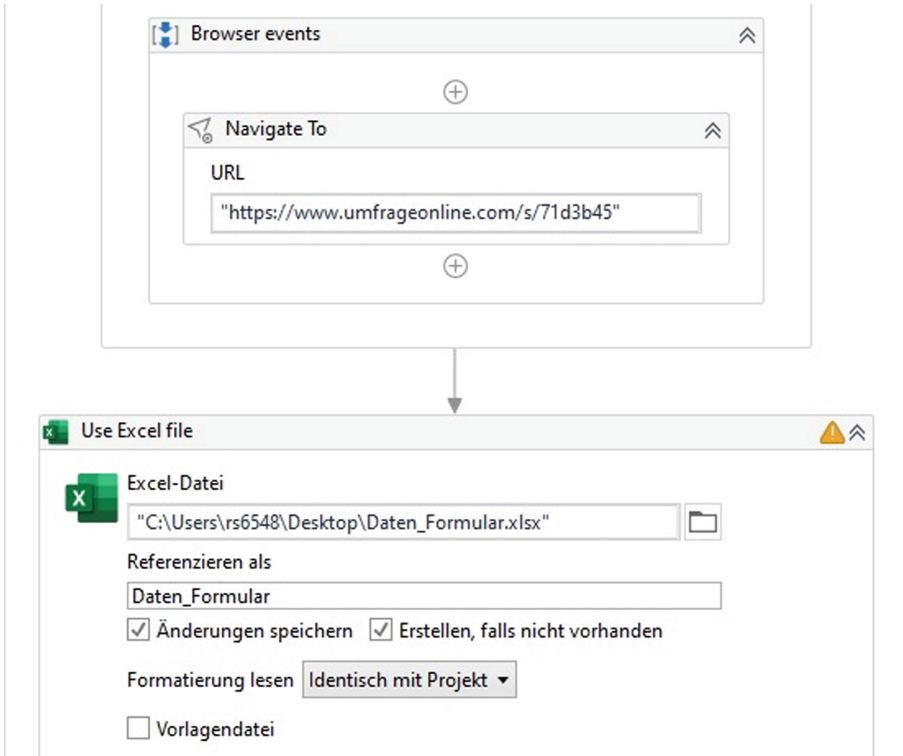


Fig. 6. Flowchart based on the generated XAML file

factor is the fact that process models might get very large. Since the data is notated directly in the activities of the process model, activities with different data are no longer mapped in one node of the model. This can lead to an explosion of the model, e.g. in the exemplary case there are 8 different activities. Due to the assignment of the explicit data to the activities, the paths in the process model have a minimum of 37 nodes. Finally, there is also potential for improvement in the graphical representation of the process models. For example, it would be useful to graphically highlight the most frequent path in the process model.

6 Conclusion

The goal of the paper at hand was the automatic creation of RPA flowcharts with Process Discovery. For this purpose, the capability of existing Process Discovery algorithms for the generation of RPA flowcharts has been analyzed. In addition, a prototype has been developed that enables this generation and thus overcomes existing shortcomings. Based on the results of the research, the prototypical implementation of a process discovery algorithm in the context of RPA could be

realized. The prototype fulfilled the desired functions in the tests and enabled the automatic generation of RPA flowcharts with required data assignments. The results also show that Process Discovery methods need to be adapted for use in the context of RPA. As the prototype showed, however, this is already possible by simply adapting the UI logs used, with all the accompanying advantages and disadvantages. The advantage is the automated generation of a process model in which the data required for automation with RPA is included. This model includes all variants of the examined process and thus enables the assessment and selection of the most suitable variant for automation.

While the prototype allowed the implementation of the functions desired, there is still room and need for improvement. Since the detection of paths in the process model is done with a rather simple algorithm, the complexity of the processes is a limiting factor. Here, the functionality could be extended to loops for example. Another interesting study would be whether the analysis of all variants of a process with Process Discovery brings great practical advantages in the context of RPA. In the prototype, two different paths can be transferred in an RPA flowchart with an if-else relation. Here, a comparison of the selection based on the analysis of all variants with the selection of the shortest or most frequent variant in the UI log and a transfer of more than two variants would be desirable.

References

1. van der Aalst, W.: Process discovery: capturing the invisible. *IEEE Comput. Intell. Mag.* **5**(1), 28–41 (2010)
2. van der Aalst, W.M.P.: Introduction. In: *Process Mining*, pp. 1–25. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-19345-3_1
3. van der Aalst, W.: Process mining: overview and opportunities. *Assoc. Comput. Mach.* **3**(2), 1–17 (2012)
4. van der Aalst, W.: *Process Mining: Data Science in Action*, vol. 2. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
5. van der Aalst, W., et al.: Process mining Manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM 2011. LNBIP*, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
6. van der Aalst, W., Bichler, M., Heinzl, A.: Robotic process automation. *Bus. Inf. Syst. Eng.* **60**(4), 269–272 (2018). <https://doi.org/10.1007/s12599-018-0542-4>
7. van der Aalst, W., et al.: Business process mining: an industrial application. *Inf. Syst.* **32**(5), 713–732 (2007)
8. van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9), 1128–1142 (2004)
9. Agaton, B., Swedberg, G.: Evaluating and Developing Methods to Assess Business Process Suitability for Robotic Process Automation. Master's thesis, Chalmers University of Technology and University of Gothenburg (2018)
10. Agostinelli, S.: Synthesis of strategies for robotic process automation. In: *Symposium on Advanced Database System*, vol. 2400, pp. 5–8. CEUR-WS.org (2019)
11. Agostinelli, S., Lupia, M., Marrella, A., Mecella, M.: Automated generation of executable RPA scripts from user interface logs. In: Asatiani, A., et al. (eds.) *BPM 2020. LNBIP*, vol. 393, pp. 116–131. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58779-6_8

12. Agostinelli, S., Marrella, A., Mecella, M.: Towards intelligent robotic process automation for BPMers. In: Proceedings of the AAAI Workshop on Intelligent Process Automation (2020)
13. Aguirre, S., Rodriguez, A.: Automation of a business process using Robotic Process Automation (RPA): a case study. In: Figueroa-García, J.C., López-Santana, E.R., Villa-Ramírez, J.L., Ferro-Escobar, R. (eds.) WEA 2017. CCIS, vol. 742, pp. 65–71. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66963-2_7
14. Barnett, G.: Robotic Process Automation: Adding to the Process Transformation Toolkit (2015). <https://www.blueprism.com/uploads/resources/whitepapers/RPA-Adding-to-the-process-automation-toolkit.pdf>
15. Bosco, A., Augusto, A., Dumas, M., La Rosa, M., Fortino, G.: Discovering automatable routines from user interaction logs. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNBIP, vol. 360, pp. 144–162. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26643-1_9
16. Buijs, J.C., Van Dongen, B.F., van der Aalst, W.: Quality dimensions in process discovery: the importance of fitness, precision, generalization and simplicity. *Int. J. Coop. Inf. Syst.* **23**(01), 1440001/1-39 (2014)
17. Callanan, E., De Venezia, R., Armstrong, V., Paredes, A., Chakraborti, T., Muise, C.: MACQ: a holistic view of model acquisition techniques. In: ICAPS 2022 Workshop on Knowledge Engineering for Planning and Scheduling (2022)
18. Cernat, M., Staicu, A.N., Stefanescu, A.: Towards automated testing of RPA implementations. In: Proceedings of the 11th ACM SIGSOFT International Workshop on Automating TEST Case Design, Selection, and Evaluation. Association for Computing Machinery, New York (2020)
19. Chakraborti, T., Agarwal, S., Khazaeni, Y., Rizk, Y., Isahagian, V.: D3BA: a tool for optimizing business processes using non-deterministic planning. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 181–193. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_14
20. FIT: PM4PY (2022). <https://pm4py.fit.fraunhofer.de/>
21. Fleischmann, A., Oppl, S., Schmidt, W., Stary, C.: Ganzheitliche Digitalisierung von Prozessen. Springer Nature, Wiesbaden (2018). <https://doi.org/10.1007/978-3-658-22648-0>
22. Fung, H.P.: Criteria, use cases and effects of information technology process automation (ITPA). *Adv. Robot. Autom.* **3**, 1–10 (2014)
23. Gao, J., van Zelst, S.J., Lu, X., van der Aalst, W.M.P.: Automated robotic process automation: a self-learning approach. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 95–112. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_6
24. Institute for Robotic Process Automation: Carnegie Mellon University: Introduction to Robotic Process Automation (2015). <https://irpaai.com/wp-content/uploads/2015/05/Robotic-Process-Automation-June2015.pdf>
25. Jimenez-Ramirez, A., Reijers, H.A., Barba, I., Del Valle, C.: A method to improve the early stages of the robotic process automation lifecycle. In: Advanced Information Systems Engineering (CAiSE), vol. 1, pp. 659–674. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_28
26. Lacity, M.C., Willcocks, L.P.: A new approach to automating services. *MIT Sloan Manage. Rev.* **58**(1), 41–49 (2016)
27. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: Colom, J.-M., Desel, J. (eds.) PETRI NETS 2013. LNCS, vol. 7927, pp. 311–329. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38697-8_17

28. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Lohmann, N., Song, M., Wohed, P. (eds.) BPM 2013. LNBP, vol. 171, pp. 66–78. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06257-0_6
29. Leno, V., Augusto, A., Dumas, M., La Rosa, M., Maggi, F.M., Polyvyanyy, A.: Identifying candidate routines for robotic process automation from unsegmented UI logs. In: ICPM, pp. 153–160 (2020)
30. Leno, V., Deviatykh, S., Polyvyanyy, A., La Rosa, M., Dumas, M., Maggi, F.M.: Robidium: automated synthesis of robotic process automation scripts from UI logs. In: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration and Resources Track at BPM, vol. 2673, pp. 102–106. CEUR-WS.org (2020)
31. Leno, V., Dumas, M., Maggi, F.M., La Rosa, M.: Multi-perspective process model discovery for robotic process automation. *Adv. Inf. Syst. Eng. (CAiSE)* **2114**, 37–45 (2018)
32. Leno, V., Polyvyanyy, A., Dumas, M., La Rosa, M., Maggi, F.M.: Robotic process mining: vision and challenges. *Bus. Inf. Syst. Eng.* **63**(3), 301–314 (2020). <https://doi.org/10.1007/s12599-020-00641-4>
33. Leno, V., Polyvyanyy, A., La Rosa, M., Dumas, M., Maggi, F.M.: Action logger: enabling process mining for robotic process automation. In: Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM, vol. 2420, pp. 124–128. CEUR-WS.org (2019)
34. Lhuer, X., Willcocks, L.: The next acronym you need to know about: RPA (robotic process automation) (2016). <https://www.mckinsey.de/business-functions/mckinsey-digital/our-insights/the-next-acronym-you-need-to-know-about-rpa>
35. Linn, C., Zimmermann, P., Werth, D.: Desktop activity mining - a new level of detail in mining business processes. In: INFORMATIK (Workshops). pp. 245–258. Köllen Druck+Verlag GmbH, Bonn (2018)
36. Slaby, J.R.: Robotic automation emerges as a threat to traditional low-cost outsourcing (2012). https://neoops.com/wp-content/uploads/2014/02/Robotic-Automation-A-Threat-To-Low-Cost-Outsourcing_HfS.pdf
37. Sutherland, C.: Framing a constitution for Robotistan (2013). https://neoops.com/wp-content/uploads/2014/03/RS_1310-Framing-a-constitution-for-Robotistan.pdf
38. UiPath: Handbook UiPath Studio (2022). <https://docs.uipath.com/studio/language/docs/flowcharts>
39. Verbeek, H., van der Aalst, W., Munoz-Gama, J.: Divide and conquer: a tool framework for supporting decomposed discovery in process mining. *Comput. J.* **60**(11), 1649–1674 (2017)
40. Weijters, A.J., van der Aalst, W., De Medeiros, A.K.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Technical report, WP 166 (2006)
41. Weijters, A.J., Ribeiro, J.T.: Flexible Heuristics Miner (FHM). In: IEEE Symposium on CIDM, pp. 310–317 (2011)
42. Wen, L., van der Aalst, W., Wang, J., Sun, J.: Mining process models with non-free-choice constructs. *Data Min. Knowl. Disc.* **15**(2), 145–180 (2007). <https://doi.org/10.1007/s10618-007-0065-y>