Multi-Perspective Process Model Discovery for Robotic Process Automation

Volodymyr Leno[1,2]
Supervisors: Marlon Dumas[1], Fabrizio Maria Maggi[1], and Marcello La Rosa[2]

[1] University of Tartu, Estonia
{leno,marlon.dumas,f.m.maggi}@ut.ee
[2] University of Melbourne, Australia
marcello.larosa@unimelb.edu.au

Abstract. Robotic Process Automation (RPA) is a novel approach for immediate cost reduction and gaining operational efficiency. RPA tools can automate repeatable tasks, thus reducing the error rates and increasing overall process performance. Even more, RPA improves the quality of the data (data completeness, data consistency/correctness, etc.). Although, being widely used in many organizations, RPA suffers from high time consumption allocated to the training of software robots (bots for short). Moreover, the models used for training are often inaccurate, which leads to increase of time spent on testing the bots. One of the possible solutions is to apply process mining in order to extract the information about the processes from UI logs such as clickstreams and keylogs, which can then be used to train the bots. However, traditional process discovery techniques are not suitable for the purpose of RPA, as they discover only control-flow perspective of the process and cannot deal well with the UI logs, producing huge and complex models. The proposed research project aims at shifting process mining techniques from working on event logs to working on UI logs as well as developing multi-perspective automated discovery technique, which can then be applied to train the RPA bots.

Keywords: Robotic process automation, Process mining, Multi-perspective model, Data-aware discovery

1 Introduction

Robotic Process Automation (RPA) [17] promises to business users to train software robots to perform repetitive, tedious, and error-prone routines in business processes, thus reducing the errors and freeing up the humans from unrewarding repetitive work so they can be reallocated to other more involving and stimulating tasks. This is achieved by providing visual WYSIWIG (what-you-see-is-what-you-get) interfaces for recording such repetitive routines in a way that they can be subsequently replicated by a bot. Moreover, automation can be extended from a task level to a business level, by also automating the handover of work between tasks. This will increase the overall process performance and

generate great cost savings. Finally, RPA improves data quality and makes data manipulation tasks more comprehensive.

While RPA has already been successfully applied to various organizations (e.g. Telefonica O2 [8], Xchanging [18]), to date, a great deal of time is still required to manually train the RPA bots, i.e. to program how they should operate. While RPA bots can be trained through a "flowchart" or software code that defines how users should interact with the particular UI, the creation of these artifacts is time-consuming and error-prone and requires a deep knowledge of the UIs involved and how users should interact with them. The negative consequences of mistakes introduced in these artifacts are magnified by the large number of bots that are typically deployed in an organization that adopts RPA. Hence, in practice, considerable time is invested in quality-testing the bots before deployment [17].

Process mining [16] is an emerging technology that has been successfully used to automatically build a range of process analytics from event logs, i.e. from the process execution data that is recorded by enterprise systems (e.g. an ERP system). A common analytic is the flowchart of the business process (i.e. the process model), that can be automatically discovered from such logs, which describes the order in which process tasks are performed within a business process. User actions on UIs (e.g. filling out a field, pressing a button) can also be recorded in the form of UI (examples of which are clickstreams and keylogs), for example by programming the software that manages the UI (e.g. a Web application server) or by using automated testing suites (e.g. SeleniumHQ). Then the process models extracted from such logs can in principle be used to train the bots.

Nowadays, there are plenty of automated discovery techniques, but most of them take only control-flow into consideration and neglect the discovery of data conditions, resource allocation, etc. A process model that covers only the flow perspective is hereby called a control-flow model, while one that covers multiple perspectives (e.g. control-flow, data, resources, time) is called multi-perspective process model. In order to create effective models for RPA, other perspectives, in particular data, apart from control-flow have to be considered as well.

Process mining techniques currently expect as input process execution data (e.g. records of process activities start and completion) whereas for RPA we need to use UI logs (clickstreams, keylogs) as input. This data is on a much lower level of granularity and process mining techniques will likely discover huge spaghetti-like models, that are not efficient for RPA. One of the possible solutions is to discover only frequent patterns and frequent partial behavior in the form of so-called local process models. However, not all the local process models are amenable for automation, therefore only those that can be automated should be discovered. In this context, the notion of "goodness" of the models has to be defined in order to pick only the ones, suitable for automation.

There are two types of approaches for modeling business processes, and hence, two classes of automated process discovery techniques. All the discovered models can be divided into procedural and declarative. The procedural models specify

all the allowed behavior and they are most suited for predictable processes in stable environments. The process is considered to be predictable when it is possible to determine with a high likelihood the path it will follow. In comparison with procedural "closed" models, i.e., all that is not explicitly specified is forbidden, declarative models are "open" and tend to offer more possibilities for execution. Instead of explicitly specifying the flow of the interactions among process activities, a declarative model describes a set of constraints that must be satisfied throughout the process execution. The possible ordering of activities are implicitly specified by constraints and anything that does not violate them is possible during execution. In this way, they are the best solution for dynamic processes characterized by high complexity and variability due to the changeability of their execution environments. In this research, both types of models are considered.

2 State of the Art

In [11], the authors propose a data-aware technique for the discovery of the declarative models. The technique uses a data-aware extension of the Declare language, which is defined in terms of LTL-FO (First Order Linear Temporal Logic). The approach is able to discover data conditions to discriminate between cases in which a constraint is satisfied and cases in which the constraint is violated. The main limitation of this technique is that only a small class of data conditions is taken into consideration. This class has to be extended, in particular, correlated conditions between activations and targets can be discovered.

Another work in the field of multi-perspective discovery of declarative models, presented in [14], proposes a mining approach that works with RelationalXES, a relational database architecture for storing event log data. Relational event data is queried with conventional SQL. Queries can be customized and cover the semantics of MP-Declare. However, the queries have to be manually specified.

In [10] a technique to mine finite state machines extended with data is presented. This work is built on top of a well-known technique to mine finite state machines that incrementally merges states based on automata equivalence notions (e.g., trace equivalence). However, this approach is not able to deal with concurrency because of the nature of automata.

One of the existing techniques for data-aware discovery of procedural models [13] is able to discover process models with conditions in the decision (a.k.a. branching) points. The approach combines existing techniques for discovering control-flow process models (e.g. Petri nets) and decision trees. The discovered conditions compare the variables with some constant values. This approach does not allow one to discover the conditions comparing two variables, or conditions involving a linear combination of the variables. A technique presented in [2] overcomes this limitation, by combining standard decision tree learning with a technique for the discovery of (likely) invariants from execution logs, i.e., Daikon [4]. The technique uses Daikon as an oracle to discover conditions that, given a decision point (e.g., XOR-split), discriminates between the cases where one

branch of the decision point is taken and those where the other branch is taken. However, to have conditions only in the branching points is not sufficient for RPA. The bot not only needs to be able to evaluate conditions at the branching points, but it also needs to relate the data manipulated by one task to the data manipulated by other tasks. In particular, the bot needs to know how the outputs of a task depend on its inputs (i.e. data transformations). This means that the discovered process model needs to relate post-conditions of a task with pre-conditions, and more generally, it needs to discover correlated conditions, i.e. conditions that relate the data available at one point in the process, with data available at earlier points in the process.

The paper [3] presents a technique to mine frequent patterns in the form of process models through a two-step approach. The method uses a concept of an instance graph - a graph representation that shows parallel and sequential steps in a trace. On the first stage, each trace from the event log is transformed into an instance graph. Then, using the set of obtained instance graphs, a graph clustering technique is applied in order to obtain frequent subgraphs. This technique can discover a limited set of constructs, such as sequential and parallel, and other constructs (e.g. choices, loops) cannot be discovered. The next work [15] extends the set of construct that can be discovered. It presents a technique to mine a set of generalizing patterns in the form of local process models. The discovered models represent frequent patterns and allow for concise summarization of the process behavior.

3 Research Problems and Research Questions

As it was shown before, a number of techniques have been proposed for the multi-perspective discovery of process models, both procedural and declarative. The problem of discovering multi-perspective process models in the declarative settings is simpler, more clearly scoped, and there is previous work in this direction [11, 14]. By contrast, the problem of data-aware discovery of procedural models has been studied less deeply. The existing techniques have many limitations and are not ready to be used in RPA. Moreover, the data that is used for RPA is on a much lower level of granularity comparing to the one that is used as input for traditional process discovery techniques, which will cause the discovery of complex spaghetti-like models.

Consequently, in this research, we aim at shifting process mining techniques from working on event logs to working on UI logs, for the sake of using process mining in RPA. We will develop the techniques for data-aware discovery of procedural and declarative process models, that can be used for training of RPA bots. The declarative models will be used to monitor the execution of bots while procedural models will be used for their training.

Accordingly, this research will aim at addressing the following research questions:

- **RQ1.** How to discover local process models from UI logs, which are amenable for automation using RPA technology?

  - **RQ1.1.** How to assess the suitability of a local process model for automation using RPA technology?
  - **RQ1.2.** How to efficiently extract from an UI log the set of local process models that are the most suitable for automation?
- **RQ2.** How to make these process models data-aware?
  - **RQ2.1.** How to discover data-aware declarative process models?
  - **RQ2.2.** How to discover data-aware procedural process models?
- **RQ3.** How to use data-aware process models discovered from UI logs to train RPA bots?

4 Research Approach

This research project aims at developing novel process mining technology to extract the flowchart of how users interact with a given UI, and using this to train and test the RPA bots automatically. By significantly reducing the time taken to program the bots, this research is expected to accelerate the adoption of RPA solutions in practice.

The project will follow a Design Science [7] research method, which is based on the creation and evaluation of a set of artifacts to study and solve the problem at hand. In our case, the artifacts are the techniques and algorithms that aims at answering the research questions specified in the previous section. Accordingly to Design Science approach, the development of a technique involves 5 steps: 1) Definition of the problem; 2) Suggestion of a solution; 3) Development of the artifacts; 4) Evaluation of the artifacts; 5) Conclusion. These steps will be followed in this research project.

For **RQ1** the research will explore the possibility of constructing the process model from frequent user behavior rather than from all user behavior observed in the UI logs, in order only to retain the most frequent way of interacting with the UIs (or most performing, based on some notion of business process performance such as cycle time or quality). In this respect, it will study the application of techniques for mining frequent patterns [6, 9] and discovery of local process models [15]. For **RQ1.1** we will define the metrics to assess the goodness of the discovered models with respect to their suitability for RPA. The corresponding analysis of RPA requirements will be conducted. The results of a literature review together with the goodness metrics will be used for **RQ1.2** in order to create a technique for discovery of local process models that are amenable for automation. The approach will be evaluated based on the real-life UI logs.

For **RQ2.1** an extensive literature review of the existing techniques for multi-perspective discovery of declarative process models will be conducted. We will start from the papers [11, 14] and then will perform the search for other relevant works following the snowballing technique. The new technique for automated data-aware process discovery will be devised based on thorough analysis of existing solutions. It will be evaluated based on synthetical and real-life logs of varied characteristics accordingly to criteria such as scalability and accuracy.

For generating the synthetic logs we will use the log generator based on MP-Declare[1]. As real-life logs we will take the ones provided for the BPI Challenges[2].

RQ2.2. As a baseline we will take the technique presented in [2] and extend it to be able to discover a broader set of rules, involving not only decision-making points but also other parts of the process. The created technique will be evaluated based on the artificial and real-life logs in the same way as for the approach for **RQ2.1**.

RQ3. First, a common format for storing UI logs will be devised, and a technique to automatically record UI logs from user actions will be implemented. Then the presented techniques for **RQ2.1** and **RQ2.2** in combination with the ones obtained for **RQ1** will be used to discover the models from such logs. The discovered models will then be used to devise a technique to automatically train the RPA bots. Finally, this technique will be extended to mine an entire process of UI-based tasks, by constructing a model of the various UIs involved in a business process, so as to fully train an RPA solution for a given business process. A systematic analysis of the features already provided by commercial RPA tools will be conducted, to understand how to best integrate the proposed solution with existing commercial RPA tools. The relevance of this research will be ensured through the evaluation of the developed solution using real-life UI logs, and the validation of its perceived impact in practice via a case study in collaboration with RPA stakeholders.

5 Preliminary Results

The research thus far has studied the problem of discovering multi-perspective declarative process models (**RQ2.1**). In the context of this research, we have developed an approach for the automated discovery of multi-perspective declarative process models able to discover conditions involving arbitrary (categorical or numeric) data attributes, which relate the occurrence of pairs of events in the log. To discover such correlation conditions, clustering techniques in conjunction with interpretable classifiers are used.

The approach is based on Declare, a declarative language of representation of business processes [12]. In particular, the multi-perspective extension of Declare, MP-Declare [1] is used. The proposed approach can be seen as a step forward with respect to the one presented in [11].

The proposed approach is shown in Fig. 1. It starts with the discovery of a set of frequent constraints. A frequent constraint is a constraint having a high number of constraint instances, i.e., pairs of events (one activation and one target) satisfying it. In addition, for each frequent constraint, also activations that cannot be associated to any target (representing a violation for the constraint) are identified. Feature vectors are extracted from the payloads of these activations and associated with a label indicating that they correspond to violations of the constraint (violation feature vectors). (Unlabeled) feature vectors are also

[1] available at `https://github.com/darksoullock/MPDeclareLogGenerator`

[2] `https://data.4tu.nl/repository/collection:event_logs_real`

extracted by combining the payloads of activations and targets of the constraint instances identified in the first phase. These feature vectors are then clustered using DBSCAN clustering [5] to find groups of targets with similar payloads. Then, these clusters are used as labels for a classification problem. These labels together with the features extracted from the activation payloads are used to generate a set of fulfillment feature vectors. Violation and fulfillment feature vectors are used to train a decision tree. This procedure allows for finding correlations between the activation payloads and the target payloads. The core part of the approach (highlighted with a blue rectangle in Fig. 1) is independent of the procedure used to identify frequent constraints and can be used in combination with other techniques for frequent constraint mining (also based on semantics that goes beyond MP-Declare).

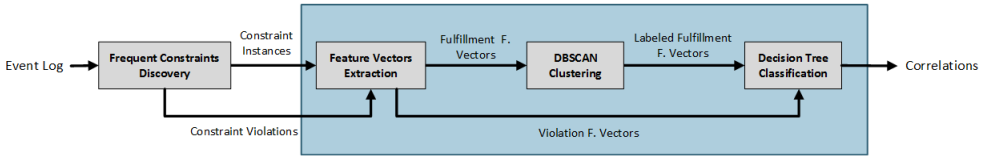

Fig. 1. Proposed approach

The approach has been validated with synthetic logs to show its ability to rediscover the artificially injected behaviors, and its scalability. In addition, the approach has been applied to 6 real-life logs in the healthcare and public administration domains in order to test the applicability of the technique in real-world settings. The results show that the approach is able to rediscover most of the constraints that generated the logs. The execution times of the technique are reasonable when the discovered models are not extremely large.

6 Conclusion and Future Work

This paper presents a research project aimed at using process mining for robotic process automation. So far, a technique for the automated discovery of data-aware declarative models has been proposed, implemented and evaluated (**RQ2.1**). This work extends the previous findings and it is able to discover much richer classes of data conditions. We still need to extend this technique in order to make it more scalable and to discover more general types of correlated conditions (e.g. conditions involving more than one variable in a term, rules involving more than two activities). Moreover, we need to evaluate these techniques on UI logs. Then we will move to the development of the artifact for **RQ2.2** and work on the **RQ1** after. Finally, we will adapt the obtained techniques for robotic process automation (**RQ3**).

References

1. Andrea Burattin, Fabrizio Maria Maggi, and Alessandro Sperduti. Conformance checking based on multi-perspective declarative process models. *Expert Syst. Appl.*, 65:194–211, 2016.
2. Massimiliano de Leoni, Marlon Dumas, and Luciano García-Bañuelos. Discovering branching conditions from business process execution logs. In Vittorio Cortellessa and Dániel Varró, editors, *Fundamental Approaches to Software Engineering - 16th International Conference, FASE 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings*, volume 7793 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2013.
3. Claudia Diamantini, Laura Genga, and Domenico Potena. Behavioral process mining for unstructured processes. *J. Intell. Inf. Syst.*, 47(1):5–32, 2016.
4. Michael D. Ernst, Jake Cockrell, William G. Griswold, and David Notkin. Dynamically discovering likely program invariants to support program evolution. *IEEE Trans. Software Eng.*, 27(2):99–123, 2001.
5. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231. AAAI Press, 1996.
6. Gang Fang, Wen Wang, Benjamin Oatley, Brian Van Ness, Michael Steinbach, and Vipin Kumar. Characterizing discriminative patterns. *CoRR*, abs/1102.4104, 2011.
7. Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.
8. Mary Lacity and Leslie P. Willcocks. Robotic process automation at telefónica O2. *MIS Quarterly Executive*, 15(1), 2016.
9. David Lo, Hong Cheng, Jiawei Han, Siau-Cheng Khoo, and Chengnian Sun. Classification of software behaviors for failure detection: a discriminative pattern mining approach. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 557–566. ACM, 2009.
10. Davide Lorenzoli, Leonardo Mariani, and Mauro Pezzè. Automatic generation of software behavioral models. In Wilhelm Schäfer, Matthew B. Dwyer, and Volker Gruhn, editors, *30th International Conference on Software Engineering (ICSE 2008), Leipzig, Germany, May 10-18, 2008*, pages 501–510. ACM, 2008.
11. Fabrizio Maria Maggi, Marlon Dumas, Luciano García-Bañuelos, and Marco Montali. Discovering data-aware declarative process models from event logs. In *Business Process Management - 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*, pages 81–96, 2013.
12. Maja Pesic, Helen Schonenberg, and Wil M. P. van der Aalst. DECLARE: full support for loosely-structured processes. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 15-19 October 2007, Annapolis, Maryland, USA*, pages 287–300. IEEE Computer Society, 2007.
13. Anne Rozinat and Wil M. P. van der Aalst. Decision mining in prom. In Schahram Dustdar, José Luiz Fiadeiro, and Amit P. Sheth, editors, *Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September*

5-7, 2006, Proceedings, volume 4102 of *Lecture Notes in Computer Science*, pages 420–425. Springer, 2006.
14. Stefan Schönig, Claudio Di Ciccio, Fabrizio Maria Maggi, and Jan Mendling. Discovery of multi-perspective declarative process models. In *Service-Oriented Computing - 14th International Conference, ICSOC 2016, Banff, AB, Canada, October 10-13, 2016, Proceedings*, pages 87–103, 2016.
15. Niek Tax and Marlon Dumas. Mining non-redundant sets of generalizing patterns from sequence databases. *CoRR*, abs/1712.04159, 2017.
16. Wil M. P. van der Aalst. *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.
17. L. Willcocks and M.C. Lacity. *Service Automation: Robots and the Future of Work*. A Steve Brookes Publishing book. Steve Brookes Publishing, 2016.
18. Leslie P. Willcocks, Mary Lacity, and Andrew Craig. Robotic process automation at xchanging. Lse research online documents on economics, London School of Economics and Political Science, LSE Library, 2015.