Author/s:
Leno, V;Deviatykh, S;Polyvyanyy, A;La Rosa, M;Dumas, M;Maggi, FM

Title:
Robidium: Automated synthesis of robotic process automation scripts from UI logs

Date:
2020-01-01

Citation:
Leno, V., Deviatykh, S., Polyvyanyy, A., La Rosa, M., Dumas, M. & Maggi, F. M. (2020). Robidium: Automated synthesis of robotic process automation scripts from UI logs. Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2020 co-located with the 18th International Conference on Business Process Management (BPM 2020), 2673, pp.102-106. CEUR Workshop Proceedings.

Persistent Link:
https://hdl.handle.net/11343/258907

Robidium: Automated Synthesis of Robotic Process Automation Scripts from UI Logs

Volodymyr Leno[1,2], Stanislav Deviatykh[2], Artem Polyvyanyy[1], Marcello La Rosa[1], Marlon Dumas[2], and Fabrizio Maria Maggi[3]

[1] The University of Melbourne, Australia
vleno@student.unimelb.edu.au, artem.polyvyanyy@unimelb.edu.au, marcello.larosa@unimelb.edu.au
[2] University of Tartu, Estonia
dvstas00@gmail.com, marlon.dumas@ut.ee
[3] Free University of Bozen-Bolzano, Italy
maggi@inf.unibz.it

Abstract. This paper presents Robidium: a tool that discovers automatable routine tasks from User Interactions (UI) logs and generates Robotic Process Automation (RPA) scripts to automate such routines. Unlike record-and-replay features provided by commercial RPA tools, Robidium may take as input an UI log that is not specifically recorded to capture a pre-identified task. Instead, the log may contain mixtures of automatable and non-automatable routines, interspersed with events that are not part of any routine as well as redundant or irrelevant events.

1 Introduction

Robotic Process Automation (RPA) allows organizations to enhance their processes by automating repetitive non-value-adding tasks. By taking over such routine tasks, RPA allows organizations to reduce errors stemming from fatigue effects and other human factors, while reducing overall cycle times [1,5].

One of the key steps in the deployment of RPA is to identify candidate tasks for automation. This step is challenging since the information about routine tasks is generally scattered across the organization. Collecting this information manually, for example via interviews, workshops, or field observations, is time-consuming and may result in un-identified automation opportunities.

In this paper, we present Robidium: a tool that aims to automate the process of routine identification and implementation. Given a User Interactions (UI) log, Robidium identifies automatable routines and synthesizes executable RPA scripts to automate these routines. This paper describes the architecture of the tool and discusses its current status and directions for enhancement. The paper is complemented by a screencast, a tutorial, and the tool itself.[4,5,6,7]

[4] Screencast available at https://youtu.be/24-pjFshquk.
[5] Tutorial available at https://github.com/volodymyrLeno/Robidium.
[6] Tool is available at http://robidium.cloud.ut.ee/.
[7] Source code: https://github.com/volodymyrLeno/Robidium (back-end) and https://github.com/stdevi/robidium-frontend (front-end).

2 Architecture

Robidium is a Software as a Service (SaaS) tool that implements the Robotic Process Mining pipeline proposed in [3]. It identifies and automates routine tasks present in UI logs. Unlike simple macro-recording tools that allow to record and replay an already well-scoped routine, Robidium discovers routines from long-running recordings of user interactions, for example a recording of a full working day. Given a UI log, Robidium proceeds by identifying recorded task instances and filtering out redundant behavior. Next, it discovers frequently repeated sequences of events (with gaps), which are then tagged as candidates for automation. Each candidate pattern is assessed for its amenability to automation. To this end, the tool discovers dependencies between data elements within each candidate and uses this information to synthesize automatable specifications. Such specifications are then compiled into executable RPA scripts.

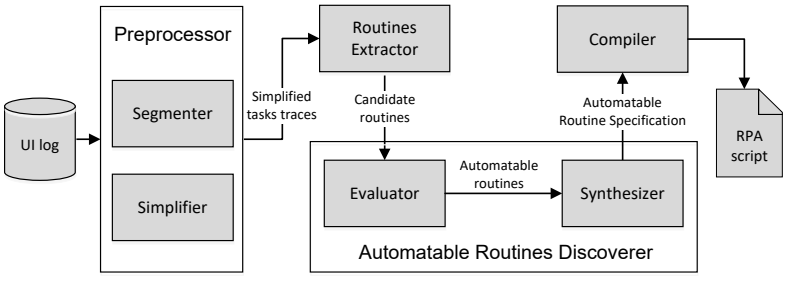


Fig. 1: Robidium architecture.

Robidium's architecture consists of six components (Fig. 1) as detailed below.
Segmenter. Robidium takes as input a UI log in which each row includes a timestamp, one or more attributes that (combined) denote an action (e.g. "Edit" + Cell ID in Excel), and other attributes capturing the action's payload (e.g. value of the Excel cell after the action). UI logs that fulfill these requirements can be produced using the Action Logger tool [4], which supports Excel and the Chrome browser. Other loggers can be used provided that they are converted to the Action Logger's format. By default, Robidium takes as input a UI log consisting of a single sequence of actions recorded during a working session. This session may contain multiple executions of one or more tasks (e.g. creating a new student record, adding a new credential to an existing student record). The *Segmenter* assumes that the user only performs one instance of one task at a time (no overlapping task instances), that the instances of multiple tasks do not share any identical actions, and that instances of multiple tasks do not always appear contiguously, but are rather separated by some events that are not part of a task instance. Under these assumptions, the *Segmenter* breaks down the single-sequence UI log into a set of sequences.
Simplifier. A UI log may contain redundant behavior that does not affect the outcome of the recorded task. For example, the user could fill in the field with

the wrong value by mistake and then correct it. This can lead to incorrect identification of routines. The *Simplifier* component eliminates such redundant subsequences of actions. It consists of three sub-modules responsible for different types of redundancies related to read, write, and navigation actions. Each sub-module is implemented via a set of regular expression find-and-replace rules.

Routines extractor. Simplifier returns a list of task traces without redundant actions. These task traces are then provided as input to *Routines extractor*, which identifies routine candidates for automation. Each user interaction in the log is converted into its symbolic representation by combining type and context attributes that capture where the action was performed (e.g., application, URL, field name, and button label). The user selects the context attributes. In order to find repeats, user interactions across the traces must have identical symbolic representations. Therefore, we do not use the attributes that contain the data used during the execution of an action (e.g., the value of a field, copied content, etc.). The tool applies sequential pattern mining to identify frequently repetitive execution patterns, given sequences comprised of symbolic representations of user interactions. Such patterns are then considered to be candidates for automation. The routine candidates can be selected accordingly to different criteria such as length, frequency, coverage, or cohesion.

Evaluator. Each candidate rutine then must be assessed for its amenability to automation by *Evaluator*. For each candidate, Evaluator extracts its instances from the log and verifies whether all the actions are automatable. In particular, an action can be automated if its value can be computed from the outcomes of the previous actions using a constant or deterministic function. In this regard, Evaluator discovers data transformations between the actions in the instances of the routine candidate. It discovers the syntactic transformations as described in [2], and semantical transformations by searching for the functional dependencies between the actions. By default, all non-edit actions (e.g., copy cell, click button) are considered to be automatable. For each routine, it then calculates a routine automatability index (RAI) as a ratio of its automatable actions.

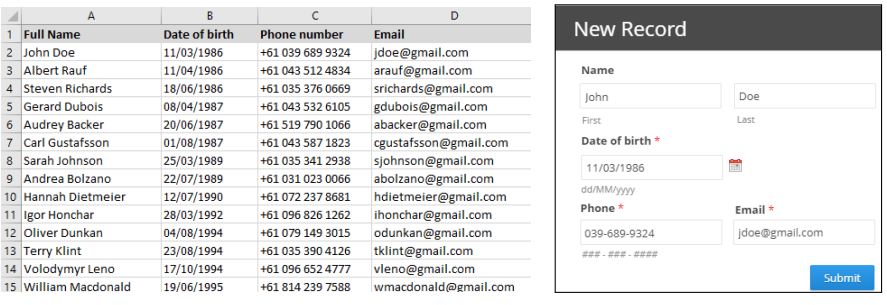
Synthesizer. Given a set of candidate routines annotated with RAI, the user can select which routine should be implemented. *Synthesizer* then prepares the automatable specification for the selected routine. It annotates the actions of routine with the corresponding data transformations and extracts the information required to map the actions to the application elements involved during routine execution (e.g., button or text field in the web form).

Compiler. This automatable routine specification is then given as an input to *Compiler* that generates an RPA bot, by mapping each action of the routine into the corresponding executable command of the selected RPA tool. At the moment, Robidium creates RPA bots for the UIPath Enterprise RPA Platform.[8] These scripts can then be executed via the command line or the interface of UIPath. Compiler also identifies the variables in the script (e.g., row in the spreadsheet) that can be then used as the input parameters during its execution.

[8] www.uipath.com

3 Example

A typical routine that can be automated using Robidium is transferring data from one system to another, for example from a spreadsheet to a form of a web-based information system. Fig. 2a shows an extract of a spreadsheet with students' contact details. Each entry in the spreadsheet is then used to create the corresponding student record using the web form shown in Fig. 2b. Such routine tasks may involve data transformations for converting the input data into the desired format (e.g., split full name into first and last name). Robidium identifies such transformations and generates corresponding RPA scripts that implement the tasks. Fig. 3 shows the generated script in the UIPath RPA platform.



	A	B	C	D
1	**Full Name**	**Date of birth**	**Phone number**	**Email**
2	John Doe	11/03/1986	+61 039 689 9324	jdoe@gmail.com
3	Albert Rauf	11/04/1986	+61 043 512 4834	arauf@gmail.com
4	Steven Richards	18/06/1986	+61 035 376 0669	srichards@gmail.com
5	Gerard Dubois	08/04/1987	+61 043 532 6105	gdubois@gmail.com
6	Audrey Backer	20/06/1987	+61 519 790 1066	abacker@gmail.com
7	Carl Gustafsson	01/08/1987	+61 043 587 1823	cgustafsson@gmail.com
8	Sarah Johnson	25/03/1989	+61 035 341 2938	sjohnson@gmail.com
9	Andrea Bolzano	22/07/1989	+61 031 023 0066	abolzano@gmail.com
10	Hannah Dietmeier	12/07/1990	+61 072 237 8681	hdietmeier@gmail.com
11	Igor Honchar	28/03/1992	+61 096 826 1262	ihonchar@gmail.com
12	Oliver Dunkan	04/08/1994	+61 079 149 3015	odunkan@gmail.com
13	Terry Klint	23/08/1994	+61 035 390 4126	tklint@gmail.com
14	Volodymyr Leno	17/10/1994	+61 096 652 4777	vleno@gmail.com
15	William Macdonald	19/06/1995	+61 814 239 7588	wmacdonald@gmail.com

(a) (b)

Fig. 2: An extract of a spreadsheet (a), and a new record web form.

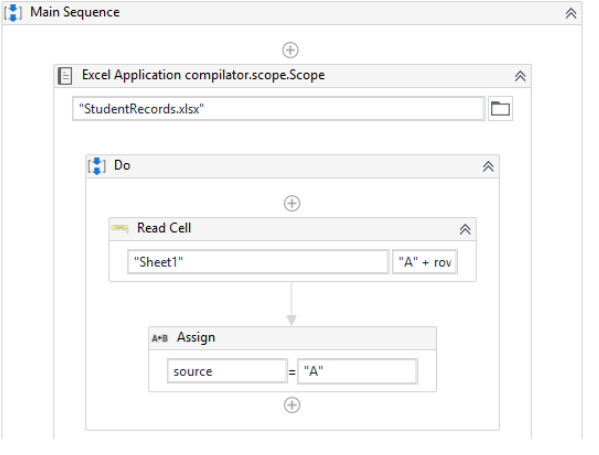


Fig. 3: A fragment of the RPA script in UIPath.

The RPA bot shown in Fig. 3 automatically transfer all the entries in the spreadsheet in Fig. 2a into the system with the web form interface from Fig. 2b.

4 Maturity

We have validated the tool in cooperation with a team responsible for admission and scholarship allocation processes within a university. The team used the Action Logger tool [4] to record daily tasks and produce UI logs that can be used as input to Robidium. We tested the components of the tool on such logs and validated the results with the workers responsible for the execution of the recorded tasks. The tool discovered several routines, and it was confirmed that they correspond to the real processes followed in the University. Since the University uses different RPA platform than the one currently supported by our tool, we could not test the Compiler component. To this end, we used the UI logs from our previous research [2]. The generated bots were able to replay the captured tasks, and they also were successfully applied to unseen data.

5 Conclusions and Future Work

This paper presented Robidium, a tool to automatically discover and implement routine tasks recorded in UI logs. The tool aims to reduce the amount of time spent on the identification and analysis of the candidates for automation and allows focusing on their implementation. The tool generates executable bots that can be used as a starting point for further refinement by RPA developers.

In future work, we plan to address some of the limitations Robidium. First, the current version of Robidum can only generate scripts of fully automatable routines. It does not support steps that require intermediate user input. Second, the tool automates only one variant of routine at a time. If a routine has multiple variants, multiple bots are generated. We plan to add functionality to combine multiple variants of a routine into a single executable specification that can be compiled into a single bot. Finally, we plan to improve the efficiency of the various algorithms implemented in the tool to be able to support larger UI logs.

Acknowledgments. This research is supported by the Australian Research Council (DP180102839) and the European Research Council (project PIX).

References

1. M. Lacity and L. Willcocks. Robotic process automation at telefónica O2. *MIS Quarterly Executive*, 15(1), 2016.
2. V. Leno, M. Dumas, M. La Rosa, F. M. Maggi, and A. Polyvyanyy. Automated discovery of data transformations for robotic process automation. In *Proc. of the AAAI Workshop on Intelligent Process Automation (IPA)*, 2020. https://arxiv.org/abs/2001.01007.
3. V. Leno, A. Polyvyanyy, M. Dumas, M. La Rosa, and F. M. Maggi. Robotic process mining: Vision and challenges. *Business & Information Systems Engineering*, 2020.
4. V. Leno, A. Polyvyanyy, M. La Rosa, M. Dumas, and F.M. Maggi. Action logger: Enabling process mining for robotic process automation. In *Proc. of the Business Process Management Demonstration Track*. CEUR, 2019.
5. L. Willcocks, M. Lacity, and A. Craig. Robotic process automation at Xchanging. Technical report, London School of Economics and Political Science, 2015.