BPM Hub: An Open Collection of UI logs

Antonio Martínez-Rojas[1], Hajo A. Reijers[2] and J.G. Enríquez[1]

[1]*University of Seville. Computer Languages and Systems Department. E.T.S. Ingeniería Informática. Avenida Reina Mercedes, s/n, 41012, Seville, Spain.*

[2]*Department of Information and Computing Sciences. Utrecht University. Princetonplein 5, 3584 CC, Utrecht.*

Abstract

A common problem in domains, such as, Robotic Process Automation or Task Mining, is to obtain event logs that include screenshots, i.e., a User Interface (UI) log. This paper presents a data package consisting of synthetic UI logs and corresponding screenshots to facilitate the testing and validation of process analysis proposals. The UI logs are generated based on an easily adaptable configuration. They closely resemble real-world use cases in the administrative domain, including three distinct processes with different activities, variants, and decision points. Multiple scenarios with slight variations in the Graphical User Interface (GUI) components are generated for each log. This data package provides a valuable resource for evaluating and refining approaches in screen-based contexts.

Keywords

UI Log, User Interface, Task Mining, Robotic Process Automation, Event Log Generation

1. Introduction

Robotic Process Automation (RPA) has gained significant attention recently as organizations seek to automate repetitive tasks and improve operational efficiency [1, 2]. RPA projects typically start with an analysis phase. For this, recent studies focus on monitoring humans performing the work and then mining the resulting event logs [3, 4, 2]. These event logs generated by the interaction between the user and the front-end of the information systems are called User Interface logs [5] (hereafter, UI logs). Several methods have been proposed for acquiring UI logs, such as Action Logger [6, 4], which captures user clicks, keystrokes, and application information, as well as the RPA logger [7], which includes screenshots in addition to clicks and keystrokes. In the process of mining these monitoring logs, instances may arise where decision points cannot be evaluated solely based on the recorded events, leading to an interest in using screenshots. Analyzing these screenshots enables the deduction of the reasons behind decision points. However, privacy concerns present challenges in obtaining publicly available UI logs with screenshots, as the screenshots may contain sensitive business data, potentially violating end-user privacy by exposing confidential information such as bank details and addresses. The scarcity of publicly available UI logs with screenshots complicates the validation of process analysis proposals [8] and the performance of tests [9] based on such logs. To overcome these

Proceedings of the Demonstration & Resources Track, Best BPM Dissertation Award, and Doctoral Consortium at BPM 2023 colocated with the 21th International Conference on Business Process Management, BPM 2023, Utrecht, The Netherlands, September 11-15, 2023

amrojas@us.es (A. Martínez-Rojas); h.a.reijers@uu.nl (H. A. Reijers); jgenriquez@us.es (J.G. Enríquez)
0000-0002-2782-9893 (A. Martínez-Rojas); 0000-0001-9634-5852 (H. A. Reijers); 0000-0002-2631-5890 (J.G. Enríquez)

challenges, we propose developing a collection of synthetic UI logs that can be easily customized by modifying the generation configuration. This synthetic log dataset will facilitate the testing and validating various proposals in the field, thereby addressing the aforementioned issues.

2. Data specification

A data package has been published[1] consisting of synthetic UI logs along with corresponding screenshots. These UI logs closely resemble real-world use cases within the administrative domain. They exhibit varying levels of complexity, measured by the number of activities, process variants, and visual features that influence the outcome of decision points. For its generation, the BPM Log Generator tool[2] [5] has been used. This tool stands out as the sole solution capable of generating screenshots for each log entry. For this a *initial generation configuration* is required:

- **Seed log**: with a single instance for each process variant and their associated screenshots.
- **Variability configuration**: to generate data from seed log. It includes two files:
  - **Case-level**: refers to variations in the content that can be introduced or modified by the user, e.g., variations in the text inputs, selectable options, checkboxes, etc.
  - **Scenario-level**: refers to vary the GUI components related to the look and feel of the different applications appearing in the process screenshots.

The data package comprises three distinct processes, P1, P2, and P3, for which their initial configuration is provided, i.e., a tuple of *<SeedLog, Case-level variability configuration, Scenario-level variability configuration>*. They are characterized by the following:

- **P1. Client Creation**: 5 activities and 2 variants. The decision point in this process revolves around the presence of an attachment in the reception of an email.
- **P2. Client Deletion. User's presence in the system**: 7 activities and 2 variants. The decision in this process is based on the result of the user's search in the Customer Management System (CRM), represented by a checkbox.
- **P3. Client Deletion. Validation of payments**: 7 activities and 4 variants. The decision involves two conditions: (1) the presence of an attachment justifying the payment of the invoices in the email and (2) the existence of pending invoices in the user CRM profile.

These problems depict processes with a single decision point, without cycles, and executed sequentially to ensure a non-interleaved execution pattern. Particularly, P3 is more complex since two visual characteristics determine its decision point.

For each problem, *case-level* variations have been applied to generate logs with different **sizes**, in the range of {10, 25, 50, 100} events. In cases where the log exceeds the desired size, the last instance is removed to maintain completeness. In addition, each log size has its associated **balanced and imbalanced** log, where balanced logs have an approximately equal distribution of instances across variants. In contrast, imbalanced logs have a frequency difference of more than 20% between the most frequent and least frequent variants. To ensure the results' reliability, 30 scenarios are generated for each tuple *<Problem, LogSize, Balanced?>*. These scenarios exhibit

[1]Available at: https://doi.org/10.5281/zenodo.8202749
[2]Available at: https://canela.lsi.us.es/bpmloggenerator/public

slight variations at the *scenario-level*, particularly in the *look and feel* and user interface of the applications depicted in the screenshots. Each scenario consists of UI logs corresponding to specific problems categorized by log size (10, 25, 50, 100) and balanced? (Balanced, Imbalanced). Here is an example from the file structure of P1. UI logs and their corresponding screenshots are organized in folders named as follows: *sc{scenarioId}_size_{LogSize}_{Balanced?}*.

```
P1/
        sc0_size_10_Balanced/            sc30_size_100_Balanced/
            screenshots/                     ...
                1_img.png                initial_generation_configuration/
                2_img.png                    seed.csv
                ...                          seed_screenshots.zip
            log.csv                          case-level_variabilty_conf.json
        sc0_size_10_Imbalanced/              scenario-level_variabilty_conf.json
            ...                          decision.json
```

Furthermore, each problem includes two additional artifacts. Firstly, the *initial_generation_configuration* folder contains the data necessary for generating problem data using the BPM Log Generator tool [5]. Secondly, the most valuable component of this package that is the "decision.json" file, which serves as the decision *oracle* for each process. This file specifies the condition that drives the decision made at the decision points. Here is an example:

```
"UI_compos": {
    "B": { // activity whose screenshot contains the UI compos
        "compo1": [499, 210, 598, 278], // bounding box: [x_left_corner, y_left_corner,
                                                x_right_corner, y_right_corner]
        "compo2": [374, 230, 482, 250]
    }
},
"decision": {
    "V1": "exists(B_compo1) or exists(B_compo2)", // condition to be variant 1
    "V2": "not(exists(B_compo1) or exists(B_compo2))" // condition to be variant 2
}
```

This testing *oracle* serves as a label for validating mined data. The file is divided in two main sections: *"UICompos"* and *"decision"*. The *"UICompos"* section includes a key for each activity related to the decision, storing key-value pairs that represent the UI components involved, along with their bounding box coordinates. The *"decision"* section defines the condition for a case to match a specific variant based on the mentioned UI components.

3. Preliminary analysis

This section is intended to provide an example of the use of this data package. Specifically, this example consists of performing a decision model discovery study using the provided data package. The analysis utilizes a decision mining tool[3], which focuses on extracting information from events preceding the decision point (DP).

Table 1 presents the relevant properties of the UI logs associated with each problem, specifically concerning events before DP. These properties include the number of variants (#variants),

[3]Available at: https://github.com/RPA-US/screenrpa

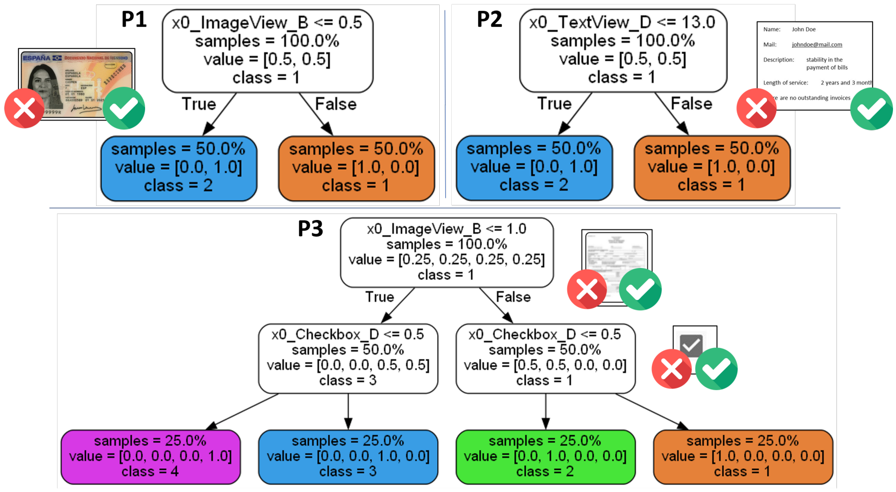

Figure 1: Initial decision model discovery results from UI logs balanced with size 100.

the average number of activities (avg #act), the number of activities occurring before the decision point (#act before DP), the average number of UI components per screenshot, i.e., UI compo. density prior to the decision point (#UICompo density before DP), and the average number of UI components influencing the decision point condition (avg #UICompo in DP).

Table 1
Preliminary analysis over the data package.

P	#variant	avg #act	#act before DP	#UICompo before DP	density	avg #UIcompo in DP
P1	2	5.0	2	24.00		2
P2	2	5.5	4	15.63		14
P3	4	5.5	4	17.13		5

Considering that the first two activities (A: view mailing list, and B: view mail) are common to all three problems. It can be observed, as expected, that the density varies depending on the nature of the activity. Where A and B are denser than the rest, since P1, which only considers these before the DP, has the maximum UI components density. The preliminary analysis, in this case, focuses on the "best" configuration where all UI components involved in the decision point are discovered, as specified in the "decision.json" file. This configuration corresponds to a **balanced** log of size **100**. The results of applying the decision model discovery approach proposed by [8] are illustrated in Figure 1. This shows a decision tree for each problem. Variables considered in the tree rules are associated with the number of UI components of a type present in an activity screenshot. For instance, in the case of P1, if there is more than one *ImageView* in activity B (there is an id card attached in the mail), the case corresponds to variant 1.

Although the differences between these problems influence their behavior under specific configurations, this analysis emphasizes the optimal case. For a more comprehensive understanding and in-depth analysis, please refer to a recent publication [8]. The data package's identifiable decision points further demonstrate its usefulness for conducting tests and validation.

4. Conclusions

This paper introduces a synthetic UI log data package with screenshots, which can be easily modified using BPM Log Generator tool [5]. The data package's key feature is the presence of clear and identifiable decision points, making it valuable for contrasting and comparing different approaches. It enables the BPM community to validate similar approaches using this data package and allows small modifications in the *initial generation configuration* to adapt the data to specific approaches. Future work includes expanding the data package to increase its variability and realism, thereby enhancing its usefulness for RPA research and development efforts.

Acknowledgments

This publication is part of the projects PID2019-105455GB-C31 and PID2022-137646OB-C31, funded by MCIN/ AEI/10.13039/501100011033/ and by the "European Union"; the FPU scholarship program, granted by the Spanish Ministry of Education and Vocational Training (FPU20/05984) and its mobility grants (EST23/00732).

References

[1] J. G. Enríquez, A. Jimenez-Ramirez, F. Dominguez-Mayo, J. Garcia-Garcia, Robotic process automation: a scientific and industrial systematic mapping study, IEEE Access 8 (2020) 39113–39129.

[2] V. Leno, A. Polyvyanyy, M. Dumas, M. La Rosa, F. M. Maggi, Robotic process mining: vision and challenges, Business & Information Systems Engineering 63 (2021) 301–314.

[3] A. Jimenez-Ramirez, H. A. Reijers, I. Barba, C. Del Valle, A method to improve the early stages of the robotic process automation lifecycle, in: Advanced Information Systems Engineering: 31st International Conference, CAiSE 2019, Springer, 2019, pp. 446–461.

[4] S. Agostinelli, M. Lupia, A. Marrella, M. Mecella, Reactive synthesis of software robots in rpa from user interface logs, Computers in Industry 142 (2022) 103721.

[5] A. Martínez Rojas, A. Jiménez Ramírez, J. González Enríquez, H. A. Reijers, A tool-supported method to generate user interface logs, in: 56th Hawaii International Conference on System Sciences (2023), pp. 5472-5481., HICSS, 2023.

[6] V. Leno, A. Polyvyanyy, M. La Rosa, M. Dumas, F. M. Maggi, Action logger: Enabling process mining for robotic process automation, in: Proceedings of Demonstration Track at: 17th International Conference BPM 2019, volume 2420, CEUR-WS, 2019, pp. 124–128.

[7] J. M. López-Carnicer, C. Del Valle, J. G. Enríquez, Towards an opensource logger for the analysis of rpa projects, in: BPM 2020 Blockchain and RPA Forum, Proceedings 18, Springer, 2020, pp. 176–184.

[8] A. Martínez-Rojas, A. Jiménez-Ramírez, J. G. Enríquez, H. A. Reijers, Analyzing variable human actions for robotic process automation, in: Business Process Management: 20th International Conference BPM 2022, Proceedings, Springer, 2022, pp. 75–90.

[9] A. Jiménez-Ramírez, J. Chacón-Montero, T. Wojdynsky, J. G. Enriquez, Automated testing in robotic process automation projects, Journal of Software: Evolution and Process (2020).