Automate It All! Revamping the Outsourcing Industry (Extended Abstract)

Antonio Martínez-Rojas[1]

[1]Department of Computer Languages and Systems, University of Seville, Avenida Reina Mercedes, s/n, 41012, Seville, Spain

Abstract

Automating repetitive tasks has long been a priority for many organizations and has been extensively studied within the field of process science. Over the last decade, Robotic Process Automation (RPA) has emerged as a highly effective method to achieve this goal. RPA enables experts to automate and integrate information systems using graphical user interfaces, offering a fast and efficient solution for repetitive task automation. Rather than constructing software robots from scratch, Robotic Process Mining (RPM) and Task Mining (TM) approaches can be used to monitor user behavior through timestamped events—such as mouse clicks and keystrokes—which are recorded in a User Interface log (UI Log) to automatically discover the underlying process model. A significant challenge in outsourcing environments, where remote virtualized systems are commonly used, is the limited information available from traditional UI logs. These logs do not capture visual context, making it difficult to identify user activities and understand decision-making processes, especially when multiple process variants exist. Existing approaches analyze the UI Log to identify underlying rules but often neglect what is displayed on the screen, resulting in an incomplete understanding of the process. To overcome these limitations, this dissertation proposes a screen-based task mining framework that enriches UI logs by incorporating visual information through screenshots and eye-tracking data captured during each interaction. This enriched log not only improves the identification of process activities but also enables the discovery of decision models, offering a more comprehensive understanding of human behavior —particularly in outsourcing contexts. By using image-processing techniques to extract relevant visual details from the screenshots, this approach extends the current capabilities of task mining, allowing for the construction of decision models that explain user choices in greater depth. These decision models are represented as decision trees, which explicitly highlight the visual elements that influence decision-making. The proposed framework has been validated through multiple case studies involving both synthetic mockups and real-life screenshots, demonstrating a high level of accuracy in capturing user decisions. The results indicate that the overall approach significantly enhances the effectiveness of task mining, revealing information previously hidden in traditional log analysis, and has the potential to revamp the outsourcing industry by improving automation applications in this type of environments.

Keywords

Task Mining, User Interface Log, Robotic Process Automation, Desktop UI Detection, UI Hierarchy, Eye Tracking, Gaze Filtering, Process Discovery Decision Model Discovery, BPO, Outsourcing

1. Introduction

In recent years, Task Mining (TM) and Robotic Process Mining (RPM) have become the first step in the pursuit of automation in business process management [1, 2]. These techniques serves to understand and improve business processes by leveraging data to discover, monitor, and optimize workflows [1]. TM/RPM focus on capturing and analyzing the detailed tasks performed by humans within processes, providing valuable insights into how processes are executed. However, despite significant advancements, challenges remain, particularly in virtualized outsourcing environments where access to client systems is limited [3].

Previous research in TM/RPM has primarily relied on user interaction data recorded in the form of UI logs, which contain timestamped events such as mouse clicks, keystrokes, and interactions with applications. While these UI logs are useful for understanding task execution, they often fail to capture the context in which these actions occur. This lack of contextual information becomes problematic when on-screen elements, such as checkboxes or input fields, influence user activities or decisions.

Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 23rd International Conference on Business Process Management (BPM 2025), Seville, Spain, August 31st to September 5th, 2025.

amrojas@us.es (A. Martínez-Rojas)
0000-0002-2782-9893 (A. Martínez-Rojas)

Several studies have explored browser or application extensions to capture additional information [2, 4, 5], such as specific spreadsheet cells and their content. However, these methods face significant limitations, particularly in the outsourcing industry, such as back-office operations or customer service tasks.These processes frequently involve handling sensitive third-party data, requiring secure connections and compliance with strict data protection regulations. Consequently, most outsourcing environments operate within virtualized systems, such as Citrix or Remote Desktop Protocol (RDP), where users interact with applications through virtualized interfaces. In these environments, the UI is often presented as a static image, disallowing direct access to system APIs or capture structured application data. Therefore, the data that can be recorded is restricted to clicks, keystrokes, and screen images, i.e. screenshots, severely limiting the effectiveness of traditional TM/RPM techniques [3].

Key contributions in the field, such as those by Agostinelli et al. [2] and Leno et al. [4], have introduced methods for capturing and automating routine tasks from structured logs. However, these approaches are often application-specific, limiting their generalizability to broader contexts, such as outsourcing industries, which may restrict the use of additional software.

To address these limitations, this research proposes a screen-based Task Mining approach that leverages enriched UI logs with screen-derived features to improve process and decision discovery. The primary goal of this research is to determine whether a screen-based Task Mining approach can effectively capture and represent process behavior in real-life outsourcing environments. To achieve this, we formulate specific research questions to guide our investigation in this context:

- (SRQ1): *Which limitations exist when extracting the UI components from desktop screenshots?*
- (SRQ2): *How can the extraction of UI components and their relationships from desktop screenshots be optimized to overcome existing limitations?*
- (SRQ3): *How can the features that capture the relevant UI elements considered by humans when making decisions be identified?*
- (SRQ4): *Can the number of features from screenshots be reduced while retaining the relevant ones?*
- (SRQ5): *Does the textual and visual features extracted from the screen improve activity identification?*
- (SRQ6): *How can the conditions that represent human decision-making be discovered?*

By addressing these research questions, this research seeks to enhance the understanding of process behavior in outsourcing environments, enabling a more comprehensive capture of visual context during user interactions, and providing valuable insights into the decision-making process, aiming to develop more effective automation solutions.

2. Background

Automation involves using technology to perform tasks with minimal human intervention, enhancing efficiency and reducing errors. In recent years, Robotic Process Automation (RPA) has become a leading technology in this field. RPA utilizes software robots, or "bots," to automate repetitive, rule-based tasks by mimicking human interactions with digital systems and software applications. These bots interact with user interfaces (UIs) to execute tasks such as data entry, processing transactions, and responding to customer inquiries.

The lifecycle of RPA follows phases similar to traditional software development, including analysis, design, development, testing, deployment, and monitoring. This research focuses on the analysis phase of RPA, where TM/RPM are the basis. TM and RPM are use in this context to capture and analyze user interactions with applications to identify automation opportunities.

A common practice in this discipline is the use of loggers for user behavior monitoring. Loggers are tools that record user interactions with applications, capturing data such as keystrokes, mouse clicks, and screen information. These interactions are stored in UI logs, which are detailed records of user actions, including timestamps, application details, and screen captures, and serve as input to TM/RPM techniques.

Additionally, eye-tracking technology enhances user behavior monitoring by capturing not only keyboard, mouse, or screen data but also the user's gaze. Eye trackers are devices that monitor and

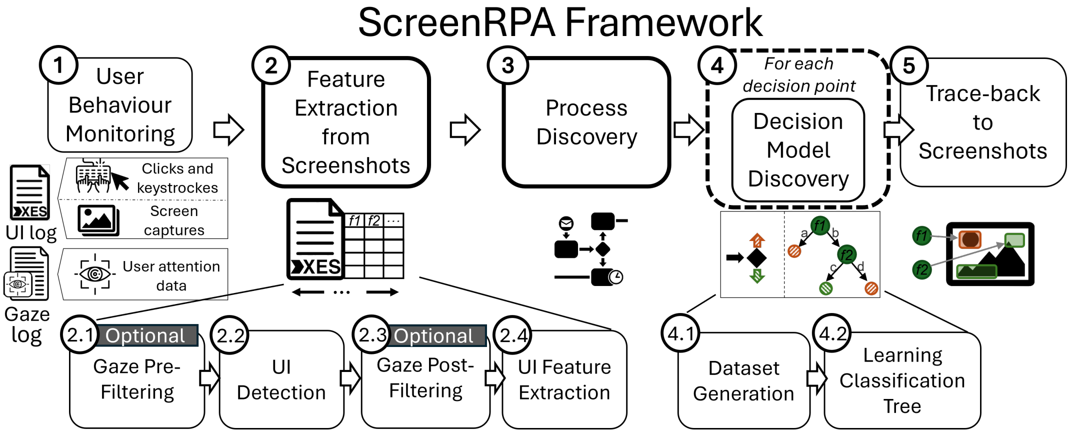
ScreenRPA Framework
1 User Behaviour Monitoring
2 Feature Extraction from Screenshots
3 Process Discovery
4 For each decision point
Decision Model Discovery
5 Trace-back to Screenshots
XES
UI log
Clicks and keystrockes
Screen captures
User attention data
Gaze log
f1 f2 ...
a
b
c
d
f1
f2
2.1 Optional Gaze Pre-Filtering
2.2 UI Detection
2.3 Optional Gaze Post-Filtering
2.4 UI Feature Extraction
4.1 Dataset Generation
4.2 Learning Classification Tree

Figure 1: Overview of the proposed framework.

analyze eye movements, providing insights into where and how long a user looks at different parts of a screen. This technology helps in understanding user attention and focus during task execution.

These concepts form the foundation for developing the ScreenRPA framework, which is described in the following section.

3. Approach

The approach of this research is structured around the ScreenRPA framework, which is designed to enhance the extraction and representation of process behavior from UI logs in real-life outsourcing settings. The framework is divided into two main phases: Enriched Behavior Monitoring and Screen-based Task Mining.

3.1. Enriched Behavior Monitoring

This phase focuses on enhancing traditional UI logs with screen-derived features to improve process and decision discovery. The key components include:

- **User Behavior Monitoring (see Fig. 1 step 1)**: Involving capturing and recording user interactions with the system. It includes logging user actions such as mouse clicks, keystrokes, and screenshots. In cases where gaze filtering is to be incorporated, it is also necessary to capture gaze data through an eye tracker. This phase gather the raw data necessary for further analysis and enrichment. The data collected here forms the basis for the subsequent steps in the framework.
- **UI Elements Detection (see Fig. 1 step 2.2)**: Utilizing a multi-model detection method based on deep learning to identify and classify UI components from screenshots. This method addresses the limitations of existing UI detection techniques by employing a hierarchical detection strategy that uses separate models for different levels of the UI hierarchy. This phase addresses SRQ1 and SRQ2 by identifying limitations in current UI detection methods and proposing a multi-model approach to optimize the extraction of UI components and their relationships.
- **Gaze Filtering (see Fig. 1 steps 2.1 and 2.3)**: Incorporating gaze tracking to filter relevant UI components based on user attention. This involves merging UI logs with gaze logs to create a unified one, so called User Behavior (UB) Log, which is then used to apply pre-filtering and post-filtering techniques to retain only the most relevant UI components. This phase addresses SRQ4 by reducing the number of features from screenshots while retaining the relevant ones.
- **Extracting Features from UI (see Fig. 1 step 2.4)**: Defining User Interface Feature Extractors (UIFEs) to transform extracted data into additional attributes that enrich the UI log. These extractors can be single UIFE, focusing on specific UI elements, or aggregate UIFE, working on

multiple UI elements simultaneously. This phase addresses SRQ3 by identifying features that capture the relevant UI elements considered by humans when making decisions.

Once the enriched UI logs are generated, they serve as the foundation for the subsequent screen-based task mining phase. The enriched logs provide a more comprehensive view of user interactions, enabling better process discovery and decision model discovery.

3.2. Screen-based Task Mining

This phase leverages the enriched UI logs to propose Task Mining techniques that utilize the additional screen-derived information. The core sections include:

- **Process Discovery (see Fig. 1 step 3)**: Identifying activities using screen-derived features from UI logs, including visual and textual information. This involves applying clustering algorithms to group events by their features, coming from the enriched UI logs. Process discovery algorithms are then used to derive the process model from these identified activities. This phase addresses SRQ5 by improving activity identification through the use of textual and visual features extracted from the screen.
- **Decision Model Discovery (see Fig. 1 step 4)**: Discovering decision models through classification techniques that explain human decision-making variability in process execution. This involves creating a labeled dataset for each decision point and training an interpretable model, such as a decision tree, to classify the decisions made by users. The decision tree is generated based on the features extracted from the enriched UI logs, allowing for a clear understanding of the conditions that lead to specific decisions. This phase addresses SRQ6 by discovering the conditions that represent human decision-making.
- **Trace-back to Screenshots (see Fig. 1 step 5)**: Linking decision points back to corresponding UI components in screenshots. The trace-back mechanism allows for a clear connection between the decision rules and the visual elements that influenced user decisions. This phase addresses SRQ6 by providing a mechanism to validate and visually associate the discovered decision rules with specific user interactions. This ensures that the framework reflect user behavior in a human-readable format, making it easier to connect decision rules to user behavior.

This approach addresses the research questions by improving activity identification and discovering decision rules, demonstrating that enriched UI logs significantly enhance both the accuracy of process discovery and the understanding of decision-making processes.

Finally, the framework provides as an output a deep analysis of the process model and decision rules, which can be used to generate a report. This report includes visual representations of the process model, highlighting the identified activities and their relationships, as well as the decision rules derived from the decision model discovery phase. The report could be considered as an As-Is Process Definition Document (PDD), providing valuable insights into the current state of the process and potential areas for automation and improvements.

4. Concluding Remarks

The ScreenRPA framework has been validated through multiple case studies involving both synthetic mockups and real-life screenshots, including real-world processes from companies operating in virtualized environments typical of the outsourcing industry. These studies demonstrate the framework's ability to extract and represent process behavior from UI logs by leveraging a screen-based approach that enriches traditional interaction data with visual and contextual cues. By integrating visual features and gaze data into the task mining process, ScreenRPA captures the situational context behind user actions, enabling the identification of process activities and the discovery of decision-making patterns that remain hidden in conventional log-based analyses. This enriched perspective significantly improves the quality of process discovery (SRQ5) and supports the derivation of interpretable decision rules

(SRQ6), providing a more comprehensive and accurate understanding of user behavior in complex digital work environments.

Despite these promising results, several limitations should be acknowledged. The accuracy of UI element detection may degrade in interfaces with high complexity, deep hierarchies, or legacy designs, limiting the framework's ability to extract meaningful features (SRQ1, SRQ2). Furthermore, processes involving dense or prolonged interactions require longer UI logs to maintain performance, as decision modeling depends on the richness and relevance of the extracted features. Complex decision scenarios with overlapping or ambiguous conditions may also impact the interpretability and precision of the resulting models. Moreover, the current evaluation covers a limited set of applications and scenarios, which may not fully reflect the heterogeneity of real-world desktop environments. These limitations underscore the need for broader validation and the creation of more diverse benchmark datasets.

Therefore, future work will focus on several key directions: (1) applying interpretability techniques to explore alternative models and assess whether they can extract decision rules using smaller amounts of data; (2) conducting more extensive evaluations in real-world environments to validate the framework across diverse and complex scenarios; (3) exploring model-to-code transformation techniques to enable the automated generation of RPA bots from the discovered process and decision models; and (4) investigating the broader potential of eye-tracking data—not only as a filtering mechanism—but also as a means to weight or validate captured traces based on indicators such as user attention, emotional state, or cognitive load.

In conclusion, ScreenRPA presents a practical and innovative approach to task mining in outsourcing contexts. It enables the extraction of process representations and decision rule mappings through a screen-based method, overcoming previous limitations in accessing virtualized systems. This process analysis provides valuable documentation and actionable insights for automation initiatives. Thus, the contributions of this research lay a solid foundation for advancing the analysis of automation projects in environments where it was previously unfeasible, improving the possibilities of automation.

Acknowledgments

This work has been supported by the grant FPU20/05984 funded by MICIU/AEI/10.13039/501100011033 and by ESF+; its mobility grants (EST23/00732 and EST24/00631); and the EQUAVEL project PID2022-137646OB-C31, funded by MICIU/AEI/10.13039/501100011033 and by ERDF, EU.

Declaration on Generative AI

The author used a generative AI tool to assist with grammar, spelling, and rephrasing. The author have reviewed and edited all AI-generated content and take full responsibility for the publication's content.

References

[1] M. Dumas, M. La Rosa, V. Leno, A. Polyvyanyy, F. M. Maggi, Robotic process mining, Process Mining Handbook (2022) 468–491.

[2] 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.

[3] A. Jiménez-Ramírez, 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, Rome, Italy, June 3–7, 2019, Proceedings 31, Springer, 2019, pp. 446–461.

[4] V. Leno, A. Augusto, M. Dumas, M. La Rosa, F. M. Maggi, A. Polyvyanyy, Discovering data transfer routines from user interaction logs, Information Systems 107 (2022) 101916.

[5] J. Gao, S. J. van Zelst, X. Lu, W. M. P. van der Aalst, Automated robotic process automation: A self-learning approach, in: On the Move to Meaningful Internet Systems: OTM 2019 Conferences: Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, Rhodes, Greece, October 21–25, 2019, Proceedings, Springer, 2019, pp. 95–112.